DISSERTATION

PROCESS-BASED DESIGN OF MULTIMEDIA ANNOTATION SYSTEMS

Zur Erlangung des akademischen Grades
eines Doktor-Ingenieur (Dr.-Ing.)

Genehmigt vom Fachbereich Informatik der
Technischen Universität Darmstadt

Eingereicht von
Dipl.-Inf. Cristian Erick Hofmann
geboren in Talca (Chile)

Referenten: Prof. Dr. techn. Dieter W. Fellner (TU Darmstadt)
Prof. Dr. Volker Wulf (Universität Siegen)

Tag der Einreichung: 04.10.2010
Tag der Disputation: 06.12.2010

Darmstadt 2010
D17

delivered by
EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

Abstract

Annotation of digital multimedia comprises a range of different application scenarios, supported media and annotation formats, and involved techniques. Accordingly, recent annotation environments provide numerous functions and editing options. This results in complexly designed user interfaces, so that human operators are disoriented with respect to task procedures and the selection of accurate tools.

In this thesis we contribute to the operability of multimedia annotation systems in several novel ways. We introduce concepts to support annotation processes, at which principles of Workflow Management are transferred. Particularly focusing on the behavior of graphical user interface components, we achieve a significant decrease of user disorientation and processing times.

- In three initial studies, we investigate multimedia annotation from two different perspectives. A *Feature-oriented Analysis of Annotation Systems* describes applied techniques and forms of processed data. Moreover, a conducted *Empirical Study* and *Literature Survey* elucidate different practices of annotation, considering case examples and proposed workflow models.
- Based on the results of the preliminary studies, we establish a *Generic Process Model of Multimedia Annotation*, summarizing identified sub-processes and tasks, their sequential procedures, applied services, as well as involved data formats.
- By a transfer into a *Formal Process Specification* we define information entities and their interrelations, constituting a basis for workflow modeling, and declaring types of data which need to be managed and processed by the technical system.
- We propose a *Reference Architecture Model*, which elucidates the structure and behavior of a process-based annotation system, also specifying interactions and interfaces between different integrated components.
- As central contribution of this thesis, we introduce a concept for *Process-driven User Assistance*. This implies visual and interactive access to a given workflow, representation of the workflow progress, and status-dependent invocation of tools.

We present results from a *User Study* conducted by means of the so-called SemAnnot framework. We implemented this novel framework based on our considerations mentioned above. In this study we show that the application of our proposed concept for process-driven user assistance leads to strongly significant improvements of the operability of multimedia annotation systems. These improvements are associated with the partial aspects efficiency, learnability, usability, process overview, and user satisfaction.

Zusammenfassung

Annotation von digitalen Multimediainhalten verbindet eine Reihe von verschiedenen Anwendungsszenarien, unterstützten Medien- und Annotationsformaten, sowie involvierten Verfahren. Demgemäß verfügen heutige Annotationsumgebungen über zahlreiche Funktionen und Bearbeitungsoptionen. Daraus resultieren unübersichtlich gestaltete Benutzungsoberflächen, so dass Anwender hinsichtlich der Aufgabenabläufe sowie der Auswahl der richtigen Werkzeuge desorientiert sind.

Diese Arbeit leistet einen Beitrag zur Verbesserung der Bedienbarkeit von Multimedia Annotationssystemen. Sie führt neue Konzepte zur Unterstützung von Annotationsprozessen ein, wobei Prinzipien aus dem Bereich Workflow Management transferiert werden. Insbesondere durch die Fokussierung auf das Verhalten von Komponenten der Benutzungsschnittstelle wird eine signifikante Reduktion von Benutzerdesorientierung und Bearbeitungszeiten erreicht.

- In drei initialen Studien wird Multimedia Annotation aus zwei Perspektiven untersucht. Eine *Funktionsorientierte Analyse von Annotationssystemen* beschreibt eingesetzte Verfahren sowie Arten von verarbeiteten Daten. Des Weiteren beleuchten eine *Empirischen Studie* und eine *Untersuchung relevanter Literatur* Praktiken des Annotierens unter Berücksichtigung von Fallbeispielen und existierenden Workflowmodellen.
- Auf Basis der aus den Studien gewonnenen Erkenntnisse wird ein *Generisches Prozessmodell für Multimedia Annotation* definiert. Dabei werden identifizierte Teilprozesse und Aufgaben, ihre Abläufe, verwendete Dienste, sowie involvierte Datenformate zusammengefasst.
- Durch die Überführung in eine *Formale Prozessspezifikation* werden Informationsentitäten und ihre Relationen definiert, und somit eine Grundlage für die Modellierung von Workflows geschafft, sowie die vom technischen System verwalteten und verarbeiteten Daten deklariert.
- Weiterhin wird *Referenzarchitekturmodell* vorgeschlagen, das den strukturellen Aufbau und das Verhalten eines prozessbasierten Annotationssystems, wie auch Interaktionen und Schnittstellen zwischen integrierten Teilkomponenten spezifiziert.
- Als zentraler Beitrag dieser Arbeit wird ein Konzept zur *Prozessgeleiteten Benutzerunterstützung* eingeführt. Dieser impliziert einen visuellen und interaktiven Zugang auf den gegebenen Workflow, die Darstellung des Arbeitsfortschritts, sowie das vom Status abhängige Aufrufen von Werkzeugen.

Abschließend werden die Ergebnisse einer *Benutzerstudie* vorgestellt, die mittels des sogenannten SemAnnot Frameworks durchgeführt wurde, das die oben genannten Gesichtspunkte realisiert. In dieser Studie wird aufgezeigt, dass der Einsatz

des vorgeschlagenen Konzepts zur prozessgeleiteten Benutzerführung zu hoch signifikanten Verbesserungen hinsichtlich der Bedienbarkeit von Multimedia Annotationssystemen führt. Diese Verbesserungen können mit den Teilaspekten Effizienz, Lernförderlichkeit, Gebrauchstauglichkeit, Prozessüberblick, und Benutzerzufriedenheit assoziiert werden.

Acknowledgements

This work originated in the course of my scholarship at the interdisciplinary postgraduate school “eLearning” at the Technische Universität Darmstadt, as well as during my occupation as research associate at the Fraunhofer Institute for Computer Graphics Research Darmstadt.

In the first instance, I would like to thank my advisor Prof. Dr. Dieter W. Fellner for his support and excellent advice. Special thanks go also to Prof. Dr. Volker Wulf for acting as co-referee. I am also grateful to Dr. Tobias Schreck, Dr. Christoph Hornung, and Dr. Stefan Göbel for advices during different periods of my time as doctoral student. Moreover, I extend my thanks to Dr. Arjan Kuijper for reviews and advices during the writing process.

I thank all advisors and doctoral students of the postgraduate school “eLearning” for all the fruitful discussions and the creative as well as exciting ambiance. Very special thanks go to Nina Hollender, my “interdisciplinary research partner”. I would also like to thank Prof. Dr. Max Mühlhäuser, Prof. Dr. Regina Bruder, and Dr. Michael Deneke for discussions and feedback on my work.

I am grateful to all my colleagues at Fraunhofer IGD A6 “3D Knowledge Worlds and Semantics Visualization” for the kindly and amicable working atmosphere, and for the support during periods of prototype implementation. In this scope, I thank the students who contributed to the implementation of the system, including Leonard Swiezinski, Aubin Tchaptchet, Uwe Boettcher, and Michael Kosiol.

Furthermore, I would like to thank Prof. Nuno Guimarães and Dr. Teresa Chambel for their advice and feedback on my work during lab visit at Universidade de Lisboa, Portugal. With respect to the conducted empirical study, I am grateful to all persons who dedicated their time. To name but a few, I thank Dr. Carmen Zahn (KMRC Tübingen), Dr. Inger Marie Dalehefte (IPN Kiel), Dr. Daniel Link (TU Darmstadt), Wilfried Rüsse (TU Darmstadt), and Dr. Jürgen Steimle (TU Darmstadt). Moreover, special thanks go to Dr. Kai Richter (SAP Research) for motivating me to pursue an academic career.

Finally, my unlimited thanks go to my family and my closest friends for all the support, love, and patience.

Contents

Chapter 1

Introduction

As means of enriching digital content by additional information, annotation techniques have found their way into multiple areas of daily use and professional fields of work. Many people read and annotate every day [BBGC01]. In this context, a study revealed that more than twenty-five percent of reading in the workplace is to be regarded as *active reading* [SPMG03], at which marks and comments are made on top or in the margins of given documents [CGG00, SPMG03].

Corresponding to these facts, annotation of digital multimedia documents covers a broad field with respect to its several purposes, objectives, and application areas. For instance, the World Wide Web comprises various platforms which enable visitors to share and annotate arbitrary multimedia content with metadata, so that they are enabled to organize this content in a structured form to facilitate later retrieval [WZY06, ZBZ$^+$08]. In addition to such simple ways of applying annotations, more complex environments are employed in various professional application fields. For different purposes and objectives, such as information retrieval, content summary and publishing, content analysis, or collaborative distributed work (cf. Section 2.1.1), recent annotation environments support practices in training and education [SBM08], edutainment [RCES09], medicines [RH07], industrial design [SGL09], or e-Commerce [RFHB09].

Furthermore, down to the present day, several techniques and approaches have been developed regarding annotation of different media formats, at which multiple forms of annotations are generated. Among supported media formats are text, graphics, audio, video, or 3D-models (cf. Section 2.1.4), and annotations may appear in terms of metadata, content, relations, or dialog acts (cf. Section 2.1.3). By analogy, beginning from the times of manual annotation of digital text documents, research on annotation has long tradition and issued a large number of projects which can be assigned to different specific subareas of annotation.

1.1 Motivation and Problem Definition

> *"First, it's difficult, time consuming, and thus expensive to manually annotate multimedia content."*
>
> Stamou et al., 2006 [SOPS06]

Although a lot of research work has already been done with regard to multimedia annotation, the operative execution still remains an elaborate, time-consuming, and hence exhausting task, especially if performed in manual manner [BEP$^+$08, BR07, SOPS06, GCCG$^+$04]. Focusing on the time factor, recent scientific projects deal with the advancement of automatic annotation approaches in order to reduce or replace human effort, which will be demonstrated in Section 2.1.4.

As opposed to such directions of research, this thesis concerns the process of annotation which, regarded as an entire operative unit that issues several concrete workflows depending on different use cases, has not been sufficiently taken into consideration [ABDF07, BEP$^+$08, HHK08]. In the following paragraphs, it will be clarified why a consideration of workflows can lead to improvements regarding the execution of annotation processes.

As pointed out in the first part of this chapter, multimedia annotation comprises several purposes, application areas, formats of media and annotations, and applied approaches and techniques. Accordingly, technical development produced a large number of different systems, which were mostly designed for a specific form of annotation [DNS03, GCCG$^+$04, Mar98]. Further development focused on more comprehensive environments in order to support different forms and tasks of annotation [ABDF07]. Hence, today's annotation environments provide a wide range of functionalities which differ significantly from each other [BR07].

As a result, respective graphical user interfaces are to be regarded as multioptional and complex due to the diversity of integrated tools and editing options. Confronted with such workspaces, especially novice operators struggle with the understanding and learning of the given toolset and the required interactions. They are disoriented with regard to their current state within the process, referring to the current task as well as already accomplished and forthcoming steps. In addition to that, they are uncertain about which available tools are to be invoked in order to accomplish the recent task [SBCO01, ZZZ07]. Accordingly, described problems arise from a lacking overview and monitoring of the process.

> *"Essentially, my guiding tech tip [...] is this: Learn it before you need it! Only obtain software with features you really need. Excessive features often leave you struggling with how to use the program rather than doing your analysis."*
>
> Spiers, J.A., 2004, on annotation-assisted video analysis [Spi04]

Example: *The process of annotation-assisted collaborative video analysis, such as described in [MJ06, PH07, SPK05], is an illustrative example in order to elucidate given problems. Tasks of the respective annotation process include (i) configuration of specific application and community settings, (ii) marking and chunking contents, (iii) classification of selected contents, (iv) generation of transcriptions, (v) writing (shared) interpretations and ratings with respect to the observed facts, (vi) commenting of co-analysts results and discourse, (vii) successive re-editing of configurations and own contributions, and (viii) publishing of results. Obviously, the process implies different annotation tools which provide facilities for administration, video segmentation and definition of marks, classification, text editing, browsing, discussion, etc.*

Process-related problems, which refer to user disorientation resulting from an insufficient overview of the (annotation) task and uncertainties at selection of appropriate tools, can be transfered to the field of *Workflow Management*. In general, Workflow Management deals with the automatic system-driven execution of (business) processes [Sch00], based on a predefined workflow model or process definition [Fau00]. A process definition comprises the logical steps (tasks), task transitions, and execution rules [CB04]. Furthermore, human and technological participants are defined which are responsible for the processing of specific tasks. Technological participants include tools, services, software components, or stand-alone applications. The general objective is to maximize the efficiency of a working process [Fau00, Hol95].

According to this, a solution of given user-specific problems in order to support the annotation process can be based on fundamental principles of Workflow Management, due to provided possibilities to define and execute a workflow, at which subtasks as well as required technological participants are explicitly supplied. Thus, the general research question of this thesis is: *"Can Workflow Management techniques improve the execution of Multimedia Annotation Processes?"*

1.2 General Objectives

The general goal of this thesis is to improve the operability of multimedia annotation systems by introducing concepts to support annotation processes, particularly intending a significant decrease of user disorientation and processing times. This goal can be associated to technological achievements, as well as to aimed benefits concerning the specified problem definition.

1.2.1 Technological Achievements

Technological solutions can be subdivided into the following sub-ordinate targets: *Automatic Workflow Execution*, *Interactive Workflow Visualization*, *Explicit Service Supply*, and *Process Definition*.

Automatic Workflow Execution. A process-based multimedia annotation system is to be enabled to execute annotation workflows in automatic manner. Hence, human operators can be provided with a seamless process procedure with respect to transitions between different successive annotation tasks. According to this, an essential prerequisite in order to realize automatic workflow execution is the capability of a system to interpret a predefined workflow specification.

Interactive Workflow Visualization. Users are to be provided with a graphical representation of the annotation workflow, enabling visual and interactive access to the process to be performed. In this context, users are to obtain process-related information about (i) all tasks included in the annotation process, (ii) the execution order and rules, (iii) the current task to perform, (iv), already accomplished tasks, and (v) the forthcoming steps. Interaction particularly concerns additional facilities for a self-motivated manual selection of tasks.

Explicit Service Supply. Regarding services as functionalities, tools, software components, applications, web services, and so forth, UI components are to be invoked or closed, depending on whether the associated service can be applied to the current task to be accomplished. In this manner, users are explicitly provided with the appropriate tools.

Process Definition. As will be revealed in Chapter 4, no “best practices” can be ascertained for multimedia annotation, which can be ascribed to the diversity of potential application scenarios and purposes. In order to support different use cases of multimedia annotation, methods are to be introduced which allow an individual specification of annotation workflows by means of existing process definition techniques.

1.2.2 Aimed Benefits

This thesis aims at providing specific benefits referring to the interactive system operation, and thus make a step forward towards a reduction of effort and processing times during annotation. In this context, expected benefits are to be associated to operability aspects, which can be subdivided into the viewpoints *Efficiency*, *Learnability*, *Usability*, *Process Overview*, and *User Satisfaction*.

Efficiency. The term *efficiency* regards the effort for task completion in relation to the quality of results, as measured by the required time. Since activities of task and tool selection are assumed by the system (unless users decide to perform a task unlike the recently pretended), the duration at task transition is expected to be reduced, which also may have an significant affect on the duration of an entire workflow run.

Learnability. A further relevant operability-related aspect refers to the the suitability of an user interface for *learning* its functions and interactions required at its operation. Additionally, this thesis concerns the requirement for learning a new workflow or task procedure. By means of automatic task execution and service supply, it is to expect that *task order* and *required tools*, as two out several points to be learned regarding system operation, are dropped. Consequently, cognitive resources might be deallocated, for instance, for learning the operation of single annotation tools.

Usability. Since targeted technological solutions intend a strict alignment between executed tasks and the applicable tools, an improvement of *usability* is expected with regard to the UI's suitability for accomplishment of the given task. Furthermore, especially founded by the visualization of a given workflow, the self-descriptiveness of the UI can be enhanced.

Process Overview. In general, it is expected that an application of the technological realizations described above leads to an improved *overview* of the annotation process to be conducted. In this case, users are supported at building a mental representation of a given annotation workflow. In particular, this might lead to a reduction of errors and uncertainties referring to task and tool selection (if manually conducted), especially in the initial phases of operation.

User Satisfaction. As a general result of improvements related to the previous viewpoints, an increase of the general *user satisfaction* can be achieved.

1.3 Contributions

This thesis regards digital multimedia annotation at the intersection of human-computer-interaction and user interface design, as well as workflow management. The overall objective is to contribute improved technological approaches for individual and collaborative knowledge work. In this context, the focus is on the specific application areas of technology-enhanced learning and construction of information infrastructures. In the following, partial contributions of this thesis are presented. Figure 1.1 illustrates how these parts are built upon each other.

Initial Studies on Multimedia Annotation. Initially, three studies on multimedia annotation were conducted, elucidating multimedia annotation from two different viewpoints. First, a conducted *Feature-oriented Analysis of Multimedia Annotation Systems* is described, which identifies functionalities, tools, and approaches offered by recent systems, as well as respective forms of processed data. In this way, a set of

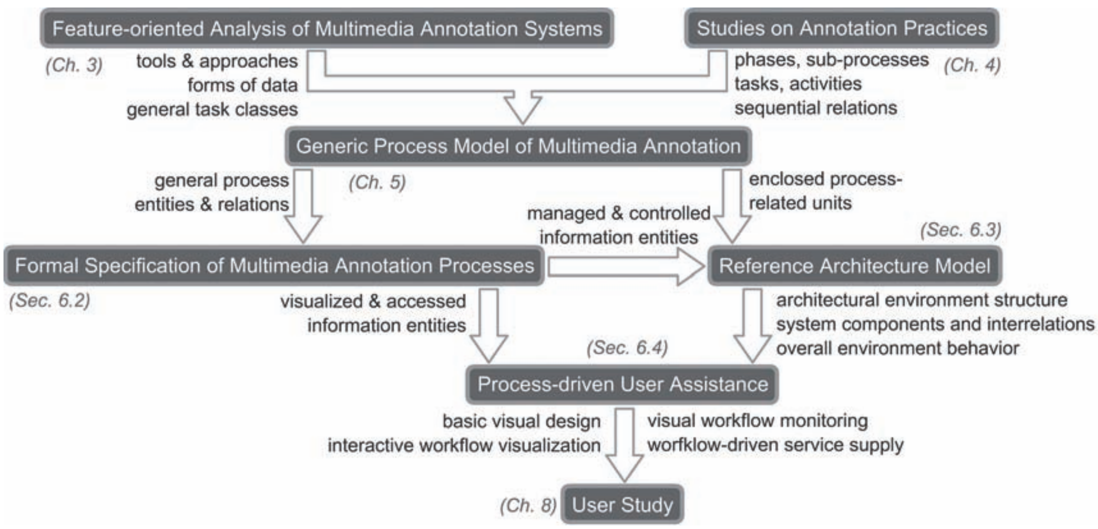


Figure 1.1: Contributions of this thesis: Context and Interrelations.

annotation features is developed and assigned to general task areas. Second, *Studies on Annotation Practices* were performed, including an *Empirical Study* and a *Survey of Related Literature*. Both studies investigate different workflows of annotation. As a result, the annotation process is subdivided into general phases, sub-processes, and tasks or activities, and relations between these items are identified with regard to their operational procedures and execution orders.

Generic Process Model of Multimedia Annotation. By means of a junction of the different results from the initial studies, a *Generic Process Model of Multimedia Annotation* is established, creating fundamental knowledge about the activities and procedures associated with multimedia annotation. The model is constructed in abstract manner, covering different use cases and workflows of multimedia annotation. It describes the contained (i) phases, sub-processes, and tasks, (ii) connecting transitions, (iii) approaches applied to accomplish specific tasks, and (vi) forms of data generated and processed by offered approaches. This constitutes the basis for a formal specification of the process, providing general entities and relations between them. Moreover, it provides process-related units which are exploited in order to define specific architecture components including respective interfaces and interactions.

Formal Specification of Multimedia Annotation Processes. A workflow scheme is specified in terms of a *Formal Specification of Annotation Processes*. Here, basic elements included in annotation processes are defined, considering specific properties and relations to other elements. In this manner, relevant classes of process-related information entities are declared, which need to be managed and by the business logic of a process-based annotation system, and are visualized and made accessible for users by respective graphical user interface. In addition to that, the formal specification serves as framework of guidelines which can be employed in the scope of workflow modeling processes, also defining general rules for workload distribution.

Reference Architecture Model. In the context of a proposed *Reference Architecture Model*, the organizational structure of process-based annotation system is determined, subdividing the overall environment into different functional sub elements. Here, a particular focus is on the consideration of workflow interpretation and execution, as well as integration and coordination of incorporated services and further building blocks. Additionally, different process-related components are introduced, defining the overall system behavior by the description of component-specific interfaces and interactions. Furthermore, an architectural solution is introduced which deals with the issue of the junction of different forms of heterogeneous data generated and processed during workflow execution.

Process-driven User Assistance. Since this thesis approaches user-specific problems which result from an insufficient consideration of the annotation process, a visual-interactive concept for *Process-driven User Assistance* is developed as central contribution. First of all, a basic visual design is described, which subdivides the user interface into four placement areas for different types of UI components. Furthermore, a model for interactive workflow visualization is established. In this context, basic graphical workflow representatives are defined, and methods for a manual selection of annotation tasks are provided. Furthermore, an approach for visual workflow progress monitoring is established. Finally, a concept for workflow-driven service supply is explained, which provides users with the right service(s) depending on the recent annotation task.

User Study. Particularly based on the implementation of *Process-driven User Assistance* as visual-interactive concept, different benefits for users of multimedia annotation systems are expected referring to the partial aspects efficiency, learnability, usability, process overview, and user satisfaction. All aspects were investigated in the scope of a user study conducted by means of SemAnnot, a realized framework and toolset for semantic multimedia annotation, which implements the concepts established in this thesis. The study particularly focused on the comparison between two application versions, with and without process assistance. For all considered aspects, the evaluation demonstrates significant improvements that are achieved by an application of process assistance. In other words, it is proofed that process-driven user assistance supports the annotation process with respect to the following points: (i) saving of time at task processing, (ii) facilitated learning of UI functions and interactions, (iii) improved usability, (iv) sophisticated overview of the process including status changes on run-time, and (v) enhancement of user satisfaction.

1.4 Outline of this Thesis

Chapter 2 provides an introduction of the theoretical fundamentals related to the subject areas *Digital Multimedia Annotation* and *Workflow Management*, particularly constituting a coherent terminology of the principal terms used in this thesis. In the first part, a general overview of multimedia annotation is given, succeeding with an elucidation of the different purposes, objectives, involved user roles, application areas, and annotation forms. Then, specific aspects of multimedia editing, collaborative work, and eLearning are illustrated. Beyond that, the second part of this chapter presents an introduction of Workflow Management, including a general definition of the related terms, and a description of workflow management systems, process definition techniques, and various workflow perspectives and patterns.

In *Chapter 3*, the conducted feature-oriented analysis of multimedia annotation systems is described. First, the underlying methodology for the identification of system features is illustrated, continuing with the exemplification of the examined applications and their detected features. Then, the incorporated systems are compared by contrasting and subsuming the essential functionalities according to different feature classes. In the last section, conclusions that particularly regard the scope of this thesis are derived.

Chapter 4 illustrates two conducted studies on annotation practices, including an empirical study and a survey on literature concerning annotation workflows. In both parts, concrete workflows of annotation are derived by describing the comprised subprocesses and more granulated tasks, as well as their sequential interrelations. Then, these use case specific workflows are compared and assembled, at which a further basis for the definition of a generic process model is constituted.

In order to establish knowledge about how persons actually work with annotation systems, *Chapter 5* presents a generic process model of multimedia annotation, which involves the partial processes, tasks, procedures, functionalities, and data forms which play a role in various concrete annotation workflows. First, the underlying methodology with respect to the model-building work is described, before elucidating the defined generic process model. Succeeding, the validity and applicability of the established model is discussed, and results comprised in the model are summarized.

Chapter 6 describes a conceptual solution framework, which is divided into three constituent parts, covering different aspects and layers of a respective technological solution. First, requirements are derived with a special focus on the specific characteristics of annotation workflows and collaborative use cases. Then, according to the distinction of different partial concepts, a formal process model is constituted. The next section addresses characteristics and components of a developed reference architecture model. Subsequently, a concept for process-driven user assistance is elucidated. A summary of the main results is given in the last part of this chapter, including a verification for the fulfillment of specified requirements by established concepts.

Chapter 7 illustrates SemAnnot, a framework and toolset for a process-driven semantic annotation of multimedia documents, which implements the solution concepts established in this thesis. In the first section, technologies applied for system implementation are described. Next, the general structure of the SemAnnot client application is elucidated. Succeeding, implemented annotation components, as well as two different annotation workflows are exemplified.

In *Chapter 8*, a conducted evaluation of the established concepts based on its realization within SemAnnot is described. Here, a special focus is on the visual-interactive concept of workflow support by process-driven user assistance. In particular, a comparison of two different application variants, with and without workflow support, is conducted. First of all, the applied evaluation method and general study design is explained, followed by the presentation of obtained results considering the different examined sub-aspects efficiency, learnability, usability, process overview, and user satisfaction. Then, the achieved benefits with respect to the support of annotation processes are discussed.

Finally, in Chapter *9*, the contributions and outcomes of this thesis are summarized. Furthermore, a prospect is provided regarding directions of future research in the field of process-based multimedia annotation.

1.5 Publications

A large part of the work presented in this thesis has been peer-reviewed and published within the following conference proceedings.

1. Cristian Hofmann, Dirk Burkhardt, Matthias Breyer, Kawa Nazemi, Christian Stab, and Dieter W. Fellner: **Towards a Workflow-based Design of Multimedia Annotation Systems**. In *Proceedings of ED-MEDIA 2010 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*. AACE, 2010, pp.1224-1233.

2. Cristian Hofmann and Dieter W. Fellner: **Supporting Collaborative Workflows of Digital Multimedia Annotation**. In *Proceedings of the 9th International Conference on Designing Cooperative Systems*. Berlin, Heidelberg, New York: Springer, 2010, pp-79-99.

3. Cristian Hofmann, Uwe Boettcher, and Dieter W. Fellner: **Change Awareness for Collaborative Video Annotation**. In *Proceedings of the 9th International Conference on Designing Cooperative Systems*. Berlin, Heidelberg, New York: Springer, 2010, pp. 101-117.

4. Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **Workflow-Based Architecture for Collaborative Video Annotation**. In *Proceedings of the 3d international*

Conference on online Communities and Social Computing: Held As Part of HCI international 2009. Berlin, Heidelberg, New York: Springer, 2009, Lecture Notes In Computer Science, vol. 5621, pp. 33-42.

5. Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **Task- and Process-related Design of Video Annotation Systems**. In *Proceedings of Mensch und Computer 2009: 9. fachübergreifende Konferenz für interaktive und kooperative Medien - Grenzenlos frei?!*. Berlin, Oldenbourg, 2009, pp. 173-182.

6. Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **Prozesse und Abläufe beim kollaborativen Wissenserwerb mittels computergestützter Videoannotation**. In *Proceedings of DeLFI 2009: 7. e-Learning Fachtagung Informatik der Gesellschaft für Informatik*. Berlin, 2009, Lecture Notes in Informatics, pp. 115-126.

7. Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **A WORKFLOW MODEL FOR COLLABORATIVE VIDEO ANNOTATION - Supporting the Workflow of Collaborative Video Annotation and Analysis performed in Educational Settings**. In *Proceedings of the International Conference on Computer Supported Education (CSEDU 09, 23-26 March 09, Lisbon, Portugal), Setúbal, Portugal*. INSTICC Press, 2009, pp. 199-204.

8. Kai Richter, Matthias Finke, Cristian Hofmann, and Dirk Balfanz, D.: **Hypervideo**. In *Encyclopedia of Multimedia Technology and Networking*, 2nd ed., Idea Group Pub., USA, 2009, pp. 641-647.

9. Nina Hollender, Cristian Hofmann, and Michael Deneke: **Principles to reduce extraneous load in web-based generative learning settings**. In *Workshop on Cognition and the Web: Information Processing, Comprehension and Learning, Granada (Spain)*, 2008, pp. 7-14.

Chapter 2

Digital Multimedia Annotations and Workflow Management

On the basis of the main objective of this thesis - an improvement of the usage of annotations systems based on the support of annotation workflows - this chapter provides an introduction of the theoretical fundamentals related to the relevant subject areas *Digital Multimedia Annotation* and *Workflow Management*. Especially those aspects related to the concepts acquired in this work will be highlighted. Additionally, a coherent terminology of the principal terms used in this thesis is constituted.

2.1 Annotation of Digital Multimedia

In order to comprehend the essential challenges of this thesis, it is important to overview the wide of digital annotations concerning its various formats, realizations, and usages. In this section, the field of digital multimedia annotation is illustrated by first presenting a general overview, and then elucidating the different purposes, objectives, and involved user roles that are tied with annotation systems (Section 2.1.1). In addition to that, this part continues with a recital and exemplification of various application areas of annotation software, as well as appearances of annotations (see Section 2.1.2 and 2.1.3). Beyond that, several media formats potentially supported by annotation systems are illustrated in Section 2.1.4. Since the application field of *Computer-supported Collaborative Learning* has been an important issue in the conducted research work (with respect to the empirical groundwork and the practical realization of the developed solution concept), this section closes with an explanation of how collaboration is assisted by annotations, and the role of annotation in eLearning scenarios (see Section 2.1.5 and 2.1.6).

2.1.1 Annotation - A General Overview

"College students always pick the dirtiest copy of a used book because they find all the helpful notes and highlights in that book."

Marshall, C.C., 1998 [Mar98]

Highlighting certain passages of a text and writing notes in the margins is a natural and common activity [Bru02, CGG00]. Suchlike activities have been practiced as long as printed text has become a mass-distribution media [Bul03]. A diary study showed that a significant percentage of work-related reading occurs in conjunction with writing, and over a quarter of these written artifacts are to be regarded as annotations [AGH$^+$98]. Hence, persons do not only read but also interact with a document and its contents [BBGC01]. In the process, they browse, summarize, organize, comment and mark on top of the documents with different objectives [Mar98].

Astonishingly, there is no uniform definition of digital annotation and its distinctness from other digital objects up to now. An investigation of the basic functions as well as a thorough agreement of a comprehensive and formal model are lacking [AF08b, BLR03]. Consequently, since existing models invariably regard quite specific usages, the idea of annotation and its management is still rudimentarily and of less general validity [ABDF07]. A general definition describes annotations as meta-information associated with a document providing an enrichment of the document [Bul03]. According to this, annotations can be regarded as means of extending arbitrary content by additional information. From a user's point of view, annotations enable individuals to associate own information with existing knowledge provided by the given document [AF08b, BR07]. In this way, a further layer of information is generated that can be referenced subsequently [BR07, CGG00]. Examples for such kind of additional personal information are explanations, references to further resources, or advices of the special relevance for marked parts [Mar97, Mar98]. In doing so, a common context is created that unifies the original and personal resources, elucidating the semantics of the annotated (primary) content [AF08b]. Thus, interpretation and understanding of the given contents by the individual recipient is enhanced [GCCG$^+$04]. In addition to that, the purpose of annotations can go beyond the scope of an individual enrichment of information resources. Annotations are also means of exchanging and sharing ideas and opinions of a collaborating community with respect to mutual accessible contents [ABDF07, AF08b, CGG00], which are visible and reusable for other persons [BR07, NCS09]. In such cases, the set of annotations forms a record of the work and communication of a specific community, and can thus be regarded as sort of community memory [CDTT04, FBT$^+$03, Mar98]. Accordingly, annotation techniques support collaborative work practices such as co-authoring, discourse, and information exchange [AF08b, CGG00, CDTT04, NKCM90].

Within the same document, different layers of annotations can coexist that need to be considered. Marshall [Mar98] distinguishes between private, shared, and public annotations, according to the aim and type of activity that is being performed. Private annotations are only visible and accessible to their authors [AF08b, LK03]. For instance, they may be hasty jottings scribbled on a text that we find irritating [Mar98]. In contrast, shared and public annotations are accessible by a certain community of users [LK03]. Shared annotations are treated by a team or sub-community, creating a collective layer [AF08b]. A public layer of annotations is visible and accessible to all users of the respective environment. Communities with access to shared and public annotations obtain different views of the information resources managed by a specific system [CDTT04, MB04].

From Paper-based to Digital Annotations

Digital annotation systems have been a collaterally evolved product of the digital era, applying the concepts and facilities of paper-based annotation to computing devices [BR07, Bul03]. The first developed environments supported markup with metainformation in order to classify contents for indexing and retrieval purposes, or to provide a simple semantic structure [FQA89]. Other early digital annotation systems allowed to edit collections of media objects parts in order to restructure the order in which the contents were rendered [DWC01]. The work of Davenport at MIT is regarded as the initial base model [DSP91, MD89]. Annotations have been extended from paper documents to any type of digital (particularly multimedia) document [BCL^{+}04]. Hence, a digital multimedia annotation can be regarded as additional digital information related to a document or parts of it (e.g., a website or contained image or a video stream) [BCL^{+}04].

In that sense, Bottoni et al. [BLR03] consider digital annotations as digital objects that are attached to other objects or parts of them. They distinguish between objects and annotations (as a specific type of digital object). A *digital object* is defined as a tuple of attribute value pairs $o = typeName((attr_1; val_1), (attr_2; val_2), ..., (attr_n; val_n))$. The type name indicates for the object category such as file, image, annotation, etc. Attributes such as author, title, or creation date are used for indexing and retrieval. Additionally, Bottoni and colleagues define *digital annotations* as objects of type annotation that are assigned to the content of one or more objects (or parts of them). Annotations obtain two main types of attributes: (i) the reference to the object(s) or object subset(s) it refers to, and (ii) a placeholder that allows the visual representation of the annotation within the annotated object.

A formal model of digital annotations is proposed by Agosti and Ferro [AF08b]. They identify significant macro-areas of annotation and provide definitions of concepts within these areas as well as their relationships. Figure 2.1 illustrates the formal model and the main annotation areas *Identification*, *Cooperation*, *Linking*, *Semantics*, and *Materialization*. Identification deals with the unique identification of an annotation and the annotated object(s), in particular with regard to the temporal constraints (cf. *synchronization* in Section 2.1.4). Cooperation elucidates annotation and their role as

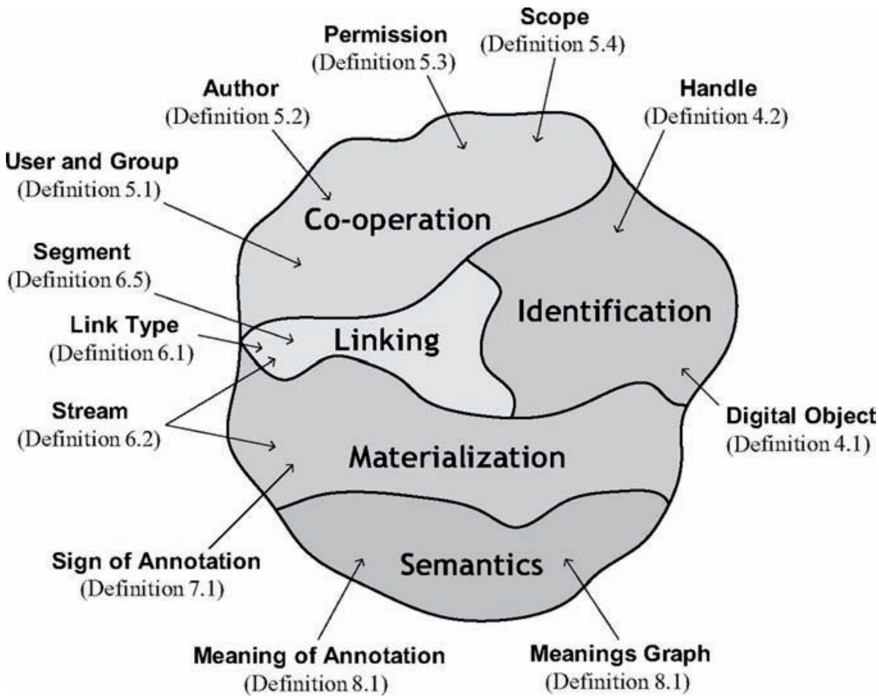


Figure 2.1: Areas covered by a Formal Model of Digital Annotation. [AF08b]

instrument for cooperation between different users (cf. Section 2.1.5). Linking highlights the problem of permitted forms of links between annotations and digital objects and, in that context, with the correct anchoring of annotation entities (cf. *anchoring* in Section 2.1.4). Semantics deals with the semantics of an annotation's content or parts of it, concerning its meaning. Finally, materialization describes the way in which annotation semantics are formed, i.e., how the content of an annotation is perceived by the user. Definitions of the concepts included in these main areas can be looked up in [AF08b].

With a particular focus on communication in collaborative use cases, Zupancic presents an abstract model of digital annotations [Zup06]. Here, an annotation as well as the annotated object may be a media object such as a text, video, image, etc. As shown in Figure 2.2, annotations are assigned to different information about the *author*, *date* of creation, and *subject* or title. Specific access rights are specified by the declaration as *private*, *shared*, or *public*. For retrieval purposes, an annotation obtains a unique *identifier*, e.g., a URI (Uniform Resource Identifier) or an ID in a database. The remaining elements of this model describe relations between annotations. These include references to its *superior annotation* and to *replies* on the annotation as well. Additionally, two types of linking elements (anchors) are related. The *context-anchor* defines the point at which the annotation is connected to the annotated digital object (cf. Section 2.1.4). A *content-anchor* describes the content of the annotation, for instance, a textual comment or a media file.

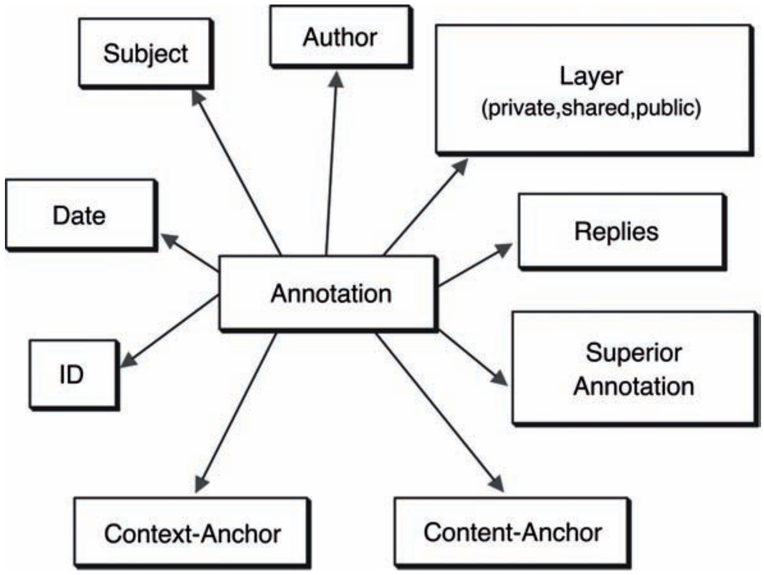


Figure 2.2: Abstract Model of (Communication-related) Annotations. [Zup06]

In fact, digital annotations are similar to annotations on printed documents. Nevertheless, they provide accessory advantages with respect to specific restrictions of paper-based annotation. Reporting on shared annotations in print cultures, Wolfe and Neuwirth [WN01] illustrate that people mostly obtain individual printouts of a text, so that any annotation they attach remains in their private context. The only way of sharing annotations are passing the document down to other readers or the formal publication channels such as explanatory footnotes. Consequently, readers of print media are limited in communication and learning through studying other persons interaction with the document. Digital technologies, especially the broad adoption of the Internet and World Wide Web provide common accessible resources, in which annotations can be stored, exchanged, and published [CGG00]. Continuative key benefits of digital annotations are their *physical boundlessness*, the possibility of *simultaneous generation*, their *availableness*, and *independency* from the original (annotated) contents [NCS09, RK03, SAYU01]. In addition to that, electronic documents, in contrast to paper documents, can be rapidly exchanged between spatial separated peers with regard to team work scenarios [Zup06]. Hence, digital annotations significantly facilitate asynchronous collaboration. An example is the *eMargo* System [GGR$^+$05], which allows collaborative annotation of digital lecture notes (see Figure 2.3).

Annotation Systems - Functions, Objectives, and User Roles

Nowadays, annotation systems comply various functions. Hanks [Han79] indicated several synonyms for the term annotation, e.g., comment, commentary, elucidation, explanation, footnote, gloss, interpretation, scholium, or jotting. In this context, fun-

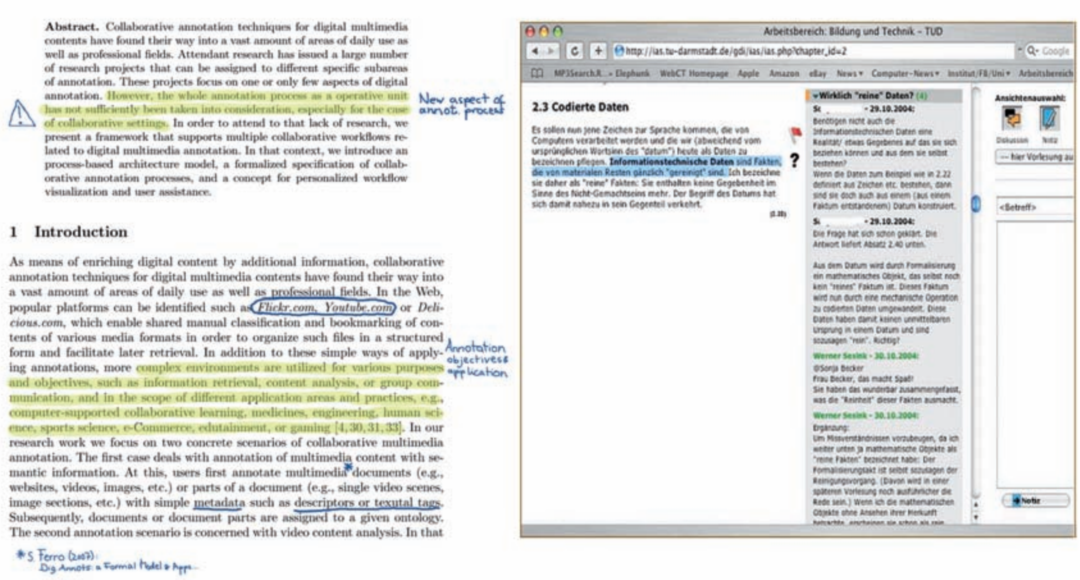


Figure 2.3: Paper-based Annotation of a printed document vs. Collaborative Annotation of digital lecture notes with *eMargo*. [GGR+05]

damental functions of annotation can be: granulated highlighting or bookmarking of document parts for memorization, establishing links, paths and networks of related information, or inserting interpretative notes that reflect the understanding of the given content [Mar97, Mar98], compare different information, save related ideas, or building activity items [SPMG03], comment on given facts of previous contents in order to establish discourse [CDTT04, MLCG07], or gathering of secondary data and categorization of contents for future content localization [Bul03].

Hence, elementary purposes and objectives of annotation can be derived such as studying or learning in general (see Section 2.1.6), or indexing and building of information infrastructures in order to improve search and retrieval [BLR03, Bul03]. In addition to that, metainformation support the generation of content summaries, e.g., for publishing purposes [OS97, MJ06]. Based on the possibility to attach objective interpretations and reflections of the given content, as well as the facilities for automatic metadata generation, annotation can be means of analysis of information which is stored in different multimedia formats [PJDM08, PH07, Rob08]. Furthermore, cooperation in the sense of task sharing, as well as collaborative treatment of the same physical data may be mediated by annotation environments [CG00, ZBM06].

According to different functions and purposes, different types of user can be identified. Bottoni et al. [BLR03] define annotation system users as a persons that access, process, annotate, and retrieve digital objects. In that context, Campanella et al. [CLM09] identify different user roles of multimedia annotation: *annotator*, *summarizer*, *media aesthetics researcher*, and *content analysis researcher*. *Annotators* work with multimedia documents and have to annotate textual or audiovisual content with metainformation and/or semantics. By means of manual or automatic techniques [LK03], *summarizer* create brief previews such as video trailers or video summaries. For that purpose, they need to rapidly navigate through the content and procure an overview of its semantic structure. *Media aesthetics researcher* want to achieve comprehension about the semantics of a professional media production. Finally, *content analyzers* interpret and reflect facts, or detect relations between low-level features and high-level concepts.

Due to the illustrated diversity of functions, objectives, and user types related to the field of annotation, today's annotation systems cover a wide spectrum of implementations and have several uses. They support multiple types of annotations that can be stored in multimedia formats and provide a multitude of functionalities differ significantly [BR07]. In the following two sections, these aspects are going to be exemplified by illustrating relevant application areas as well as forms and appearances of annotation.

2.1.2 Application Areas of Multimedia Annotation

This section presents current uses of annotations by giving selected examples of relevant application areas. Relying on a classification derived by Agosti et al. [ABDF07], usages in *The Web*, *Digital Libraries*, *Databases*, and *Search & Retrieval* are presented, enriching their work with an up-to-date view. In addition to that, the fields of annotation-aided *Augmented Reality* and *Media Content Analysis* will be illustrated.

The Web

From its initiation in the 1960s to now, the Internet has gone through a wide expansion. Primarily used by researchers at first instance, it emerged to a public network for millions of users, especially due to the adoption of the World Wide Web (WWW). The WWW exploits the Internet's global interconnectedness to distribute multimedia documents such as text, images, videos, and music [CGG00, Fin05]. Web annotations are connected to these kind of web resources. Thus, as previously illustrated in a more general context, web annotations can be related to any kind of content provided by the Web as commentary, explanations, references, and other forms of notation. Here, annotations constitute a layer that is superior to the primary resources, at which the annotated content is not physically modified [ABDF07]. Web annotation can also be aided by automatic approaches in order to reduce human efforts. In that scope, respective research work either deals with the extraction of metainformation from the resource's underlying databases, or the automatic analysis of text within web pages through learning techniques or Natural Language Processing [WZY06]. Web-based annotation systems provide features for highlighting text within a web page, adding notes at certain spaces of a document, or generating annotated hyperlinks to further resources [RK03]. The variety ranges from simple applications such as personal bookmarking of web pages by means of locally stored annotations, or more complex uses in collaborative settings supported by client-server environment [RK03]. In that context, the Web provides new opportunities for sharing multiple forms of annotations. One can publish content through the Web and thus make it accessible for friends or colleagues via web browsers [CGG00]. Furthermore, complex annotation environments can be configurated for distributed, asynchronous collaboration [CGG00]. In the following, specific applications of annotations on the Web - *Social Annotations*, *Semantic Web*, and *Recommender Systems*, are elucidated.

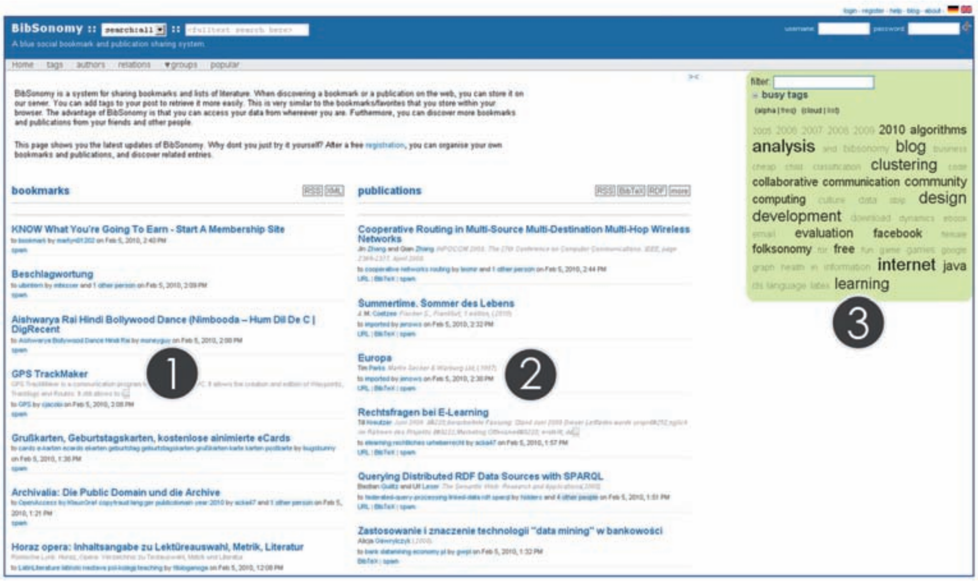


Figure 2.4: The Social Bookmark Platform *Bibsonomy* [Bib]. (1) Bookmarked web resources. (2) Incorporated publications. (3) Tag Cloud (relevance-oriented visualization of defined tags).

In the course of the recent evolution of the World Wide Web that is often abstracted as *Web 2.0*, so called *Social Bookmark* services have become popular and useful services [ZBZ+08]. Such services enable users to organize web resources by means of specific annotated metadata, and to share these content classifications with members of the same community [WZY06]. Widespread examples are *Delicious* [Del] and *Bibsonomy* [Bib] (see Figure 2.4). *Delicious* allows a personal keyword-based categorization of websites, storage of annotated categories in order to make this personal collection available at any place, and sharing of collections with other members. *Bibsonomy* supports researchers during information-seeking processes by allowing organization and sharing of scientific literature. The characteristic feature of social bookmark services is the application of *tags* (keywords freely chosen by users) that are treated as annotation units. Collections of suchlike user-created categorical tags and vocabularies have been named *folksonomy*. Mathes [Mat04] ascribes the success and benefits to basic reasons:

First, tagging does not expect sophisticated knowledge about a certain taxonomy so that folksonomies provide low entry barriers. Second, annotators obtain immediate feedback, since they are able to to see further contents that were annotated by others using the same tag(s).

Another relevant application of annotations can be localized in the area of the *Semantic Web*, which makes use of so called semantic annotations. In that case, annotations are regarded as structured, computer-understandable data that is used to enhance human-understandable data (such as text on a web page). In this manner, information can obtain well-defined meaning and automatic processing is possible [BLHL01]. The field of semantic annotation is elucidated more explicitly in Section 2.1.3.

Annotated metadata is also a fundamental aspect in the field of *Recommender Systems* [GCCG+04] (cf. *Information Retrieval*). In general, Recommender Systems are intelligent applications that assist users in finding products, services, and information in e-commerce and other environments that provide rich content [FB08, MR09]. Originally, recommender systems were developed as social systems which allowed users to share preferences and ratings over simple products such as books [FB08]. Here, classical two-dimensional approaches considering users and items have been applied [ASST05]. These approaches can be divided into content-based, collaborative, and hybrid recommendation methods [ASST05]. Current recommender systems are able to process more complex information through new approaches such as constraint-based recommendation [FB08], the exploitation of user context information [ASST05], or conversational recommendation that brings human-system interaction into focus [MR09].

Digital Libraries

Annotation is an essential instrument for providing and supporting the different functionalities of a Digital Library Management System (DLMS) [ABDF07, AF08b]. Digital Libraries not only realize the facilities that are enabled by common “real-life” libraries or archives, they also provide more than just the access to stored digital contents. The basic features of a DLMS are the creation, supply, and management of digital content. Accordingly, creation of new information through annotations is enabled in multiple ways. Any new annotation that is attached to existing information resources becomes a new information resource itself [FHM+01]. In doing so, annotations enlarge information resources provided by the Digital Library. Annotations enable users to merge and link personal information with the existing contents, so that new relationships between objects are established, creating a common context of different information [ABDF07]. Thus, the provided information space is augmented by an additional structural and semantical layer that comprises various browsing paths and an alternative structuring of the content. At the same time, annotations also can connect a hypertext or hyperspace to a DLMS. That allows users to interact with contents managed by different DLMSs in an active and dynamic way [AF05b, AFPT06].

Hence, Digital Libraries can not only be regarded as isolated environments, but also as systems that cooperate in order to improve user experience [AF08a]. In addition to that, annotation can be means of supporting the addition of helpful subjective contributions. Related ideas or opinions can be discussed, and this (digitalized) discourse can in turn be linked to respective information objects [ABDF07, AF08b]. At the same time, new annotations may contain interpretations of given content, providing additional information. In that case, users are supported with respect to the understanding of presented facts [ABDF07]. A DLMS can also apply automatic annotations, alleviating the user's first approach to the document. For example, topic detection can be used to assign annotations to their specific topics. This establishes an annotation-specific context, which enables the reorganization of the document and a fine-grained segmentation into topics [ABDF07].

Digital Libraries also afford the transfer of contents across global networks. In this manner, an effective treatment of information by large user groups is enabled [FHM$^+$01]. In this context, sharing digital contents and related annotations assists collaborative work settings, in which group members obtain common access to the same digital resources. As stated by Marshall and Ruotolo [MR02], retrieving content, reading, and integrating new annotations can be done simultaneously together with other tasks, such as working with colleagues. Thus, Digital Libraries are to be regarded as user-centered systems and "a common vehicle by which everyone will access, discuss, evaluate, and enhance information of all forms", since content management is connected to further communication and cooperation tasks [IMA$^+$05].

Figure 2.5: User Interface of *Probado-3D* and *Probado-Music*. [Pro]

One recent example is the *PROBADO* Project. Probado is concerned with the support of the lifecycle workflow for non-textual media from creation, retrieval and delivery, to maintenance [Pro]. By means of the two current applications *Probado-3D* and *Probado-Music*, access is provided to sophisticated libraries of 3D-Models and music tracks. Figure 2.5 illustrates the Graphical User Interface of both implementations.

Database Management Systems

Annotations are also applied in Database Management Systems (DBMS), for example in the context of scientific databases [ABDF07]. As such, they take an essential role as means of understanding and curating databases [EAE+09]. Annotations, especially in the form of semantic information, are embedded within the database and merged with the managed contents [BKTT04, Sci91]. For this purpose, they can be connected to database entities at different levels such as tables, tuples, columns, or cells [EAE+09]. In scientific databases, different versions of the stored information can be managed and archived over time [ABDF07, BKTT04]. The hierarchical structure of scientific data is exploited in order to represent the different versions by means of tree structures[BKTT04]. Moreover, historical information about an entity is stored [Sci91]. In this process, metainformation that includes time-stamps and structure information is assigned to single nodes of the tree. This information is related to the annotations embedded in the database schema, so that they are to be regarded as information about the changes in the set of content over time [BKTT04]. Thus, the annotated tree structure represents an additional information layer that provides facilities for archiving and searching different versions of the information managed by a DBMS.

Information Retrieval

As mentioned in Section 2.1.1, one relevant purpose of annotation is the classification of specific media contents for indexing and retrieval, as well as the gathering of metainformation for later content localization [Bul03, RH05]. In general, Information Retrieval (IR) includes finding of contents and documents of unstructured nature, broader than supported by traditional database searching [MRS08]. Annotations provide sophisticated search facilities, so that the retrieved information is improved by better rankings and more relevant search results in response to a user query [AF08b].

First generation systems supported a textual markup of contents. Furthermore, by the use of media processing techniques such as pattern recognition, media contents can be annotated not only establishing a conceptual level using keywords, but also constructing a perceptual level. For instance, Colombo et al. [CBP99] report on the retrieval of visual information from image and video data. They present an approach of enriching visual data with semantic annotations. Frommholz et al. create a discourse context by extending documents by annotation threads. Suchlike annotations are applied in order to search and retrieve documents by comparing the annotations as well as their position in the discourse and their type [FBT+03, FTK04]. Agosti and Ferro [AF05a, AF06] suggest the use of Hypertext Information Retrieval (HIR) approaches and link fusion techniques in order to provide better search strategies. Cabanac et al. [CCCJ08] present an architecture that improves IR at the levels of searching and browsing by supporting annotation practices with respect to the document lifecycle.

Here, searching is improved by considering collective annotations (user contributions) as "social feedback" to enhance recalls by obtaining more results relevant to a query. Browsing is supported by recommending documents depending on the current navigation state, also exploiting previously annotated data. Social annotations (see *The Web* in this section) are exploited in [ZBZ+08], aiming at an encouragement of the user experience in IR. A framework is presented that combines topic analysis with language modeling methods.

Although these approaches have a great potential impact in information retrieval, a factual break through is still not achieved. Further research is needed in order to make annotation really useful for information retrieval. Alonso and Zaragoza [AZ08] summarize three key challenges for future research:

- Definition and evaluation of fundamental annotation tasks as scientific work with respect to general theories. In this context, tasks have to be general enough to be assigned to the whole community and, at the same time, specific enough to be applicable. Additionally, it has to be clarified how suchlike tasks may improve competition and reproducibility.
- It is still a common problem to provide access to large sets of annotated data. Reasons are the proprietary of formats (especially in the scope of research projects), the difficult provision of identical versions of online data sets that change quickly, or the fact that small groups need to use lots of different tools for preprocessing, annotation, and indexing.
- With respect to the latter point, each project requires various technologies from different disciplines such as natural language processing, web mining, semantic web, etc. Thus, it is important to foster exchange of knowledge and resources between respective communities.

Augmented Reality

Annotation, regarded as additional attached information to a (virtual) object, is one of the most exploited concepts by Augmented Reality (AR) [WDH09]. In contrast to physical or offline information resources, AR presents required information and the related object within the same display, so that a context is provided that helps users to engage and understand (see Figure 2.6). Contextualization and localization of information are the purposes for the application of annotations. Hence, AR annotations are powerful means of presenting extra information about the world a user is moving within, conveying the information in different ways. Examples of annotation-aided AR environments are interactive guide systems, in which users are allowed to move around a certain place (city, campus, museum, etc.) and obtain information about metro-access, building labeling, or artistic installations. Wither et al. [WDH09] provide a detailed insight into the field of AR annotations.

Figure 2.6: Augmented Reality Annotations in first person view mode. [WDH09]

Media Content Analysis

In the scope of the research work presented in this thesis, also the application of annotation systems for Media Content Analysis purposes has been investigated, especially in the context of *video analysis* in collaborative eLearning scenarios. There, annotation techniques provide referable multimedia documents that serve as means of description, documentation, and evidence of analytic results [HHK08, MJ06]. In the process, annotation systems are able to provide specific functionalities for the different phases and tasks within the analysis process, such as chunking contents into relevant parts, classification of these marked contents by means of metadata, writing interpretations of the given information, creating agreement with co-analysts by adding contributions to a discussion, and publishing results [MJ06, PH07, SPK05, HHF09d, HHF09a]. Examples for application scenarios for (collaborative) video analyses in education are movie analysis in the film sciences, motion analyses in sports and physical education, or the acquisition of soft skills such as presentation or argumentation techniques [PLR06, HHD08]. A further exemplification of this application field is given in Section 4.1, describing the investigated concrete use cases.

2.1.3 Appearances of Multimedia Annotations

Various appearances of annotations can be identified that indicate for the annotation's role in its specific context. The online video sharing platform *Youtube* [You] constitutes a simple representative example. Here, textual comments and video content can be attached to primary video, and metadata is annotated, e.g., in order to recommend similar videos. Figure 2.7 illustrates the different types of annotations included by Youtube.

With respect to their specific function und purpose, annotations may have different appearances. For example, Marshall [Mar98] distinguishes between formal and informal annotations. Here, formal annotations are regarded as metadata entities that follow a certain structure, whereas informal annotations are "marginalia of the sort that we write to ourselves as we read a journal article", referring to a certain content [Mar98]. In that scope, Agosti et al. [ABDF07] broaden this view by defining annotations that may appear as hypertext and dialog acts. Accordingly, these different appearances of annotations as *metadata*, *content*, *hyperspace elements*[1], and *dialog acts* are exemplified in the following.

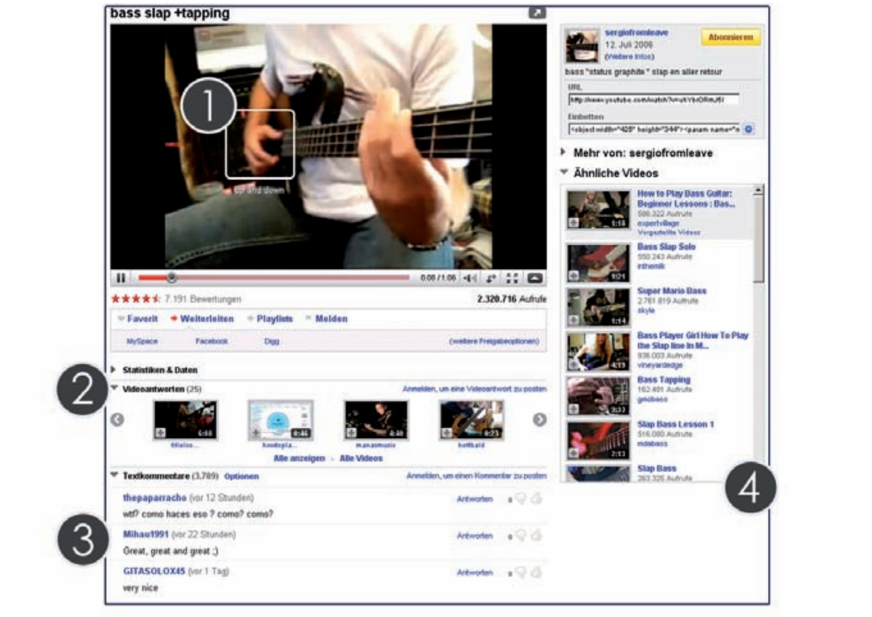


Figure 2.7: Different Forms of Annotations on *Youtube* [You]. (1) On-screen comments that may be associated to a highlighted cutout. (2) Attach further videos in order to reply to the primary content. (3) Communication and discussion through a communication board. (4) Recommendation of similar videos, based on descriptive metadata.

Annotations as Metadata

Research in the area of media annotation was initialized in the course of the development of new technologies such as media editing [FQA89]. There, annotations are applied as metainformation that relate to existing media content in order to improve content classification or to provide information about the content's properties and semantics [AF08b, Bul03, GCCG$^+$04, KK01]. This kind of additional metadata needs to adhere specifications that refer to the structure, syntax, semantics, and also values that can be assigned to annotations [AF08b]. According to this, different classes of metainformation can be localized, from text fields that are integrated in a file header [PJDM08], descriptors that provide specific information about audiovisual data [Sik01], and to shared metadata that are used as bookmarks in order to organize and manage data in a collaborative manner [KK01]. Furthermore, annotations may

[1]As the term hypertext is restricted to textual content, the term hyperspace will be used instead.

be part of a structured scheme such as a vocabulary, a taxonomy, or a category system, but can also be specified in an unstructured way, e.g. as part of a tagging system [BFW07, GRS04, SPK05]. In general, annotated metadata can be exploited by persons and computing devices as well [AF08b, CDTT04]. On the one hand, users are provided with well-structured information about the given media content. On the other hand, computer systems are allowed to automatically process annotated data sets.

Several standardization activities have led to specific formats for metadata-enriched multimedia objects. These standards enable the storage of descriptions that are to be assigned to different levels of information, e.g., a descriptive level for information specification, a structural level that supports navigation and presentation, or a configurative level for management purposes [SOPS06]. One of the most frequently applied metadata annotation format is *MPEG-7* [Sik01], which was defined by the MPEG-7 Multimedia Description Language work item of the Moving Picture Experts Group (MPEG) [MKP02, Sik01]. It offers different functionalities for description, editing, and management of multimedia content that are based on content analysis and processing [ATP$^+$05, MKP02, Sik01]. Furthermore, a main focus is the realization of efficient content-based retrieval specifications [Sik01]. As one key benefit in contrast to other formats, MPEG-7 is generic and applicable to a broad range of application domains [MKP02]. The *Synchronized Multimedia Integration Language* (SMIL) initiated the class of open XML-based standards specified by the World Wide Web Consortium (W3C) [WWWCb]. It supports time-based interactions between integrated multimedia objects, and can be applied in the context of systems that support a synchronized presentation of multimedia in time [ZB07]. The main features of SMIL are (i) extensibility with respect to language extensions, (ii) a separated storage of annotations and the “raw data”, (iii) its declarative characteristics that allow a temporal and spatial specification of the timing and layout constructs, as well as (iv) the support of complex timing, layout, linking, and content control [Bul03]. *Dublin Core*, defined within the ISO Standard 15836-2003, is a simple and flexible metadata format that provides a compact set of predefined descriptors [HR05, SOPS06]. It aims to describe a variety of different resources, and is accordingly used in different domains such as art, science, education, or business [HR05, SOPS06]. A more recent development, *MPEG-21*, brings the exchange of digital content into focus and aims at providing a normative framework for multimedia transaction and consumption [SOPS06, TAC$^+$08]. In doing so, a transparent use of media content across different networks, devices, and user types is supported [TAC$^+$08].

An advanced example for metadata annotations are *Semantic Annotations* comprised by the *Semantic Web* [AF08b, PJDM08] (cf. Section 2.1.2). The emphasis is the extraction of semantic metadata that supports content description in a conceptual level [ATP$^+$05]. As mentioned before, the general objective is to make content and its description machine-understandable, so that computing devices are enabled to process content and to foster automation, integration, and information reuse [BLHL01, HS03, Rei06]. Relevant standards for semantic annotation serialization are RDF and OWL that, in addition to SMIL, have also been specified by the W3C [WWWCa, WWWCc]. In general, these specification provide facilities to define formal descriptions, using the XML syntax.

An overview of the relations between "simple" descriptive and structural metadata and semantic annotations is given by Stamou et al. [SOPS06]. They distinguish between three abstraction levels within metadata-annotated content: a subsymbolic, a symbolic, and a logical level (see Figure 2.8). The subsymbolic layer comprises the raw multimedia data which is represented in usually binary formats that are applied for compression and streaming purposes. In the middle layer, metadata formats such as MPEG-7 are used to describe the content, e.g., defining its internal structure. Thus, a structural layer is placed on top of the media stream. Finally, the logical abstraction level provides the semantics for the symbolic layer by assigning the structured information to the specific domain knowledge representation.

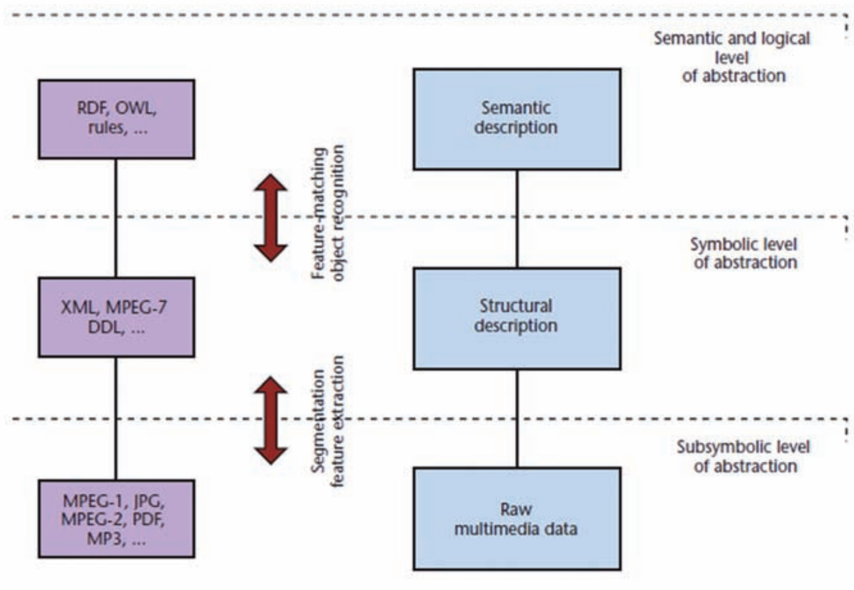


Figure 2.8: Levels of multimedia information. [SOPS06]

Annotations as Content

Unlike the previously exemplified view of annotations as metadata, annotations can also be regarded as additional content that is assigned to existing basic multimedia objects [GCCG$^+$04]. This form can be compared to active reading, at which persons add own comments in form of interpretations or ideas [ABDF07]. A popular example for these kind of annotations has been illustrated at the description of *Youtube* [You] above, where shared videos can be commented via free text (within the video display or in a related forum at the bottom) or even by means of replying videos. This implies that content-type annotations enrich the existing information, creating further layers of explanation, interpretation, and elucidation [AF08b]. Whereas metadata annotations provide elucidation through certain constrained or formal descriptions of the basic semantics, content annotations are the explanation themselves and help users to understand the given information. Nevertheless, without a layer of structural metadata, the additional information is not explicit for a machine, making it difficult to realize effective search and recommendation [GCCG$^+$04]. Consequently, content-type annotations are only interpretable for humans, since computing devices fail to automatically process data from the additional layer. In contrast to metadata, the semantics first have to be extracted by some means, before it can be processed and interpreted by the computer. On the other hand, content annotations offer additional semantics that arise from objective comments and interpretations, so that complex context and coherences, such as hidden facets, are made easier to understand by the recipient [AF08b].

Two different viewpoints are to be distinguished with respect to content-based annotations: they may either serve as mere content enrichment, or they may constitute a stand-alone document [AF03]. In the first case, annotations are additional objects assigned to already existing content within a document. They are not autonomous, but still can rely on existing resources ensuring justification. Regarded as stand-alone documents, annotations are also "real" document entities that are connected to the base resources and can be accessed independently.

Annotations as Hyperspace Elements

As already stated, annotations may support the creation of relationships between contents by means of linking, at what the existing contents and their associated annotations establish a hyperspace [AF05b, AF06] or, integrated into existing information structures, take part of it [AFFT04, Mar98].

The recipients of such a hyperspace are engaged in an active manner [Mar98]. The information space is enlarged by information addition, since users reflect and respond to given contents through comments, or create paths and connections to external resources. Besides, annotated information resources provide additional facilities for navigation and browsing, as well as advanced search and retrieval [AF08b]. For that reason, Marshall [Mar98] considers the process of annotation as a essential aspect of hypertext construction.

Annotations as Dialog Acts

As part of a collaborative discourse that is referred to a certain subject, especially content-type annotations that are connected to specific links can be regarded as dialog acts [FBT$^+$03, FTK04]. In this context, they are means of exchanging ideas between different readers and authors in relation to commonly used documents or parts of it

[FFM04]. This kind of shared communication sets comprise additional context, as well as contradictory opinions about the content or annotations [AF08b]. In doing so, not only the statement or phrased idea, but also the capability to integrate a document into the discourse it belongs to, is relevant. Thus, with respect to the aspects mentioned in the previous paragraph, annotations are means of creating an information space that contains hyperlinked communication contributions. Frommholz et al. [FTK04] define different forms of so-called *discourse structure relations* that refer to the elaboration of new information, description of similarities or contrasts between information entities, description of reasons for certain facts, giving extra background information, interpretation of statements, or the support of argumentation. Among others, the support of communication in collaborative settings is explained later in Section 2.1.5.

2.1.4 Specific Aspects of Digital Multimedia

In this section, multimedia-related aspects of digital annotations are elucidated, beginning with an examination of the term multimedia. Based on these definitions, the principles of *anchoring* and *synchronization* are presented, explaining their specific relevance with respect to the annotation of multiple media formats. Moreover, different media formats of annotations on the one hand, and the annotated content on the other hand are illustrated, giving a brief insight into recent topics of digital media annotation research.

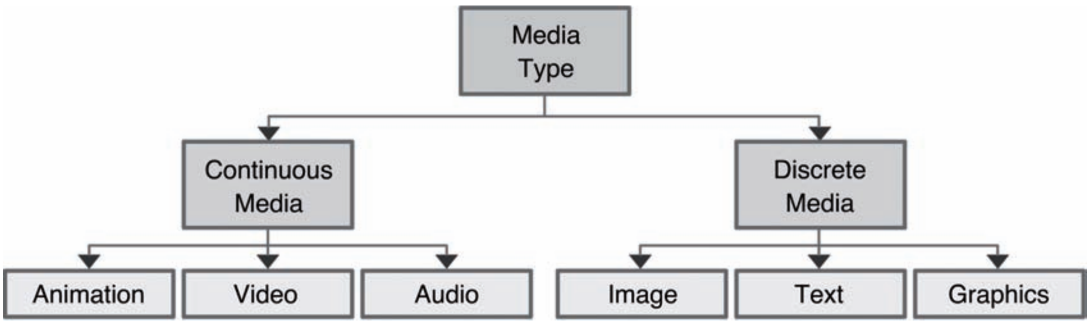


Figure 2.9: Taxonomy of media formats. [Fin05]

Definitions of Multimedia

In general, multimedia is regarded as a composition of different media such as text, audio, image, graphics, or video. Figure 2.9 illustrates a simplified taxonomy of media types that can be part of a superordinate multimedia environment. In this context, Finke [Fin05] first divides multimedia into the two general categories *continuous* and *discrete media*. Continuous media are time-dependent formats such as video, audio, and animation. The second category, discrete media, comprises the static media formats text, images, and graphics. This is a restricted view on different media encodings as some formats are missing. Nevertheless, this classification serves as basis for the comprehension of the definitions of multimedia used in this work.

A common notation of the term multimedia is lacking in literature. The following definitions all focus on different aspects, such as the temporal properties of involved media objects, the human-system interaction, and the synchronization of pooled media objects. One frequently referenced definition with regard to multimedia systems is given by Steinmetz [Ste95]:

Definition 1. *A multimedia system is characterized by a computer-driven and integrated creation, manipulation, representation, storage, and communication of independent information, which at least are coded in one continuous and one discrete medium.*

Schulmeister [Sch07] reviews this definition, stating that combinations of only time-dependent or only time-independent media (audio-video or text-image) could also be regarded as multimedia. In his definition, he emphasizes the interaction facilities as the fundamental element. At the same time, he describes multimedia content as figurative knowledge that is not interpreted until accessed by the user. In that sense, Schulmeister defines multimedia as follows:

Definition 2. *Multimedia is an interactive form of dealing with figurative knowledge within a computer-aided interaction.*

An earlier definition is presented by Naik [Nai96], emphasizing the importance of synchronization at the content presentation level:

Definition 3. *Multimedia systems are characterized by computer-controlled generation, manipulation, storage, communication, and presentation of independent media data. Synchronization among the various media data at the presentation level is key to a multimedia system making an impact on the users.*

Anchoring and Synchronization as Challenges for Multimedia Annotation

To sum up the definitions presented above, multimedia can generally be regarded as a compendium of differing media formats. Multimedia objects often refer to other (maybe not-digital) artifacts, so that a four-way relationship can be defined between the physical artifact, its annotation, the digital media object, and its annotation [GVOH05]. According to the definitions of multimedia presented above, it becomes clear that multimedia annotation systems need to deal with multimedia rich annotations, and need to process all annotations of a specific media set [AF08b, GVOH05]. These sets are divided into several parts that are coded in different media formats that require different levels of description. Thus, an important topic within multimedia annotation research is *anchoring*, dealing with the validity space and/or period of annotations. That is, when annotating digital documents, validity areas have to be specified in order to define the scope of annotations within the document [BCL+04]. For example, a whole Web document may be annotated. Beyond that, within a Web document a single text, an image, a video, or compositions of more objects can be selected. Proceeding with a further dimension, parts of such an object, e.g., spatio-temporal areas within a video, can be defined as areas to be annotated [CDTT04, CCG02, Kip08]. According to these aspects, validity areas within media contents can be classified as follows (see also Figure 2.10):

- *Multimedia Container.* Multiple media objects that establish one single entity. Examples are websites or digital presentation slides that combine text, images, and videos.
- *Media Object.* A delimited media file, such as a text or an image, that is coded in one particular media format.
- *Temporal Area.* A time point or time interval of a continuous media like audio or video.
- *Spatial Area.* A spatial delimited area within discrete media files, e.g., a paragraph within a text, coordinates within a graphic, or polygons of a 3D-Model.
- *Spatio-Temporal Area.* A field within a continuous media that features spatial and temporal properties, for instance, an object shown in a movie that is tracked and framed by a visual box over time.

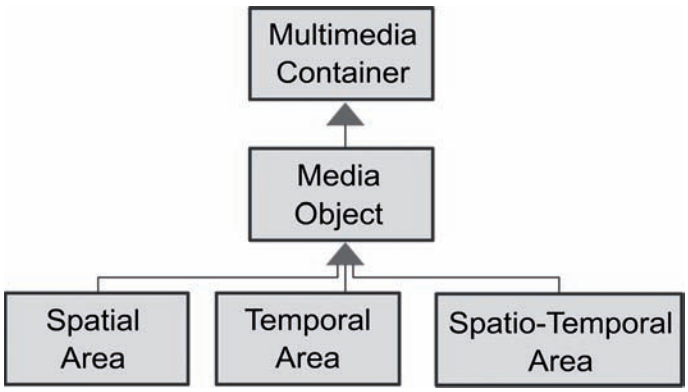


Figure 2.10: Validity Areas of Multimedia Annotations.

Closely tied to annotation anchoring is the *synchronization* of the different media objects contained in a multimedia document, as well as the objects' annotations (which, as already pointed out, also can be a media object). In general, synchronization considers the spatial, temporal, or logical relationships between involved entities. From the viewpoint of process communication and scheduling, multimedia synchronization is primarily associated to the temporal characteristics of a mixed media environment [Nai96]. Regarding the conceptualization of the graphical user interface of a multimedia display system, users must be enabled to control the presentation of connected contents interactively by pausing, resuming, or jumping to a random position [BHB+97, Nai96]. In particular, synchronization is achieved by managing events in scheduling data sets, and storing presentation as well [BHB+97]. When replaying annotations, the schedule is interpreted and the involved events are triggered. General information about multimedia synchronization can be gathered at the *W3C Synchronized Multimedia website* [WWWCb].

Concluding: *Anchoring* and *synchronization* are two relevant aspects of multimedia annotation and are of importance in this thesis. As will be discussed in Section 2.2.4, the so-called *Data Flow* needs to be specified and controlled within the annotation process. Accordingly, anchoring defines the specific structural characteristics of data that is transfered between successive tasks or tools as input or output, while synchronization specifies the relationships (especially temporal ones). As a result, workflow management for annotation systems requires the common management and junction of respective annotated information that is attached to different media objects in the scope of a multi-step process.

Recent Research on Digital Media Annotation

In digital annotation, different media formats are involved in digital annotation referring to the contents that are annotated on the one hand and the annotations themselves on the other hand [AF08b, BLR03, Fin05, Rei06]. This is the case for both continuous and discrete media types. Finally, there is an ongoing large amount of research concerning annotation of several media formats such as *Textual Media*, *Images and Graphics*, *Music and Speech*, *Video-based Media*, or *3D-Models*. In this context, Schroeter et al. [SHN07] propose a classification of annotation systems according to (i) the annotation level from free textual to semantic ontology-based, (ii) the media type, and (iii) the number of files that can be compared and interlinked simultaneously. This classification is illustrated in Figure 2.11 (where at that time state-of-the-art tools are exemplified).

A low-level survey in recent research concerning digital media annotation (for instance by submitting search queries in research literature databases) reveals that there are strong tendencies towards automatic annotation and media analysis approaches, following the common intention to improve media retrieval processes. While such approaches mostly aim at generating simple forms of descriptive metadata, the semantic level of metainformation is also a further actual topic in focus. In this scope, some examples of current research lines are presented.

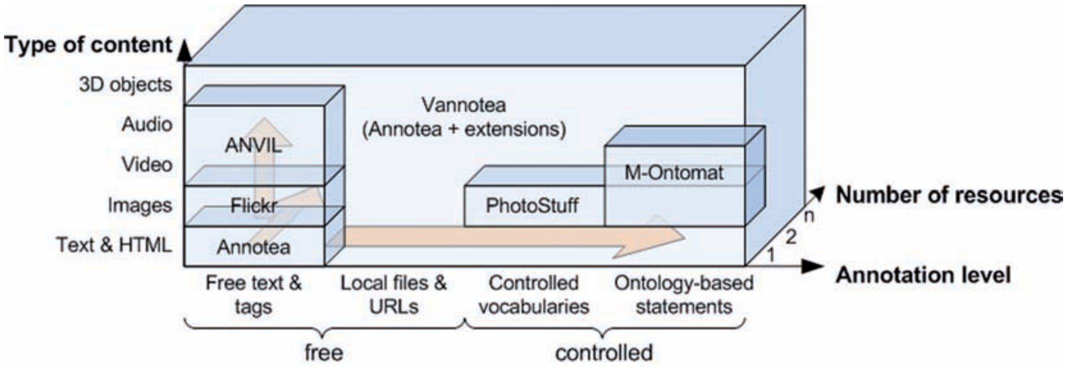


Figure 2.11: Classification of Annotation Tools. [SHN07]

Annotation of Textual Media. In terms of the general problem of tedious and time consuming manual annotation, current research addresses annotation of textual media by applying approaches of natural language processing (NLP). One central topic - transformation of raw text into structured constructs - is described in [BCEB09]. Here, free-text annotations are exploited in order to automatically infer the semantic properties of online documents. As noticed by Tomanek and Olsson [TO09], high-quality annotated data constantly includes a bottleneck in supervised machine learning methods for NLP. Accordingly, their work deals with an active learning approach for automatic annotation. NLP also plays a role in the scope of Semantic Annotation. In [DSKT+09], a (web-)service-oriented framework is presented that automatically constructs semantic word classes using semantically similar training samples. Related research exploits the logical structure of textual documents in order to assign semantic information [EBHR07, LSG+07]. For instance, using a specific domain ontology, El-Beltagy et al. [EBHR07] annotate segments within information rich text defined with the aid of on headings that are coupled with the hierarchical structure of the text. Another currently active field is automatic emotion detection based on textual data, for instance investigated by Lu et al. [LLL+10]. With the aim of fostering intelligent computing, a knowledgebase-independent approach is proposed that automatically recognizes events within sentences by means of mutual action histograms between the subject and the object of an event.

Annotation of Images and Graphics. There is an increasing number of applications that allow automatic or manual annotation of image files. A main problem is the semantic understanding of content, so that a sophisticated automatic annotation is needed that supports the reflection of personal memory (as main contextual information) and the interoperability between different systems [TTP07]. Feng and Xu [FX10] describe that, from a machine learning perspective, annotation fits to multi-instance and multi-label learning, since images are particularly specified by multiple semantic labels that are often related to image segments (instead of the whole image). They propose a transductive multi-instance and multi-label framework, addressing the annotation problem by exploiting both labeled and not-labeled data. In order to improve image recognition, learning approaches are applied to enable systems to store large amounts of targets [NHK09]. However, a bottleneck can be identified with respect to visual knowledge acquisition using search-based methods. Nakayama et al. [NHK09] approach this problem by deploying a similarity measure approach exploiting context information that is gathered from multiple labels. In the scope of image annotation by means of computer vision and image processing technologies, Bouguila and ElGuebaly [BE10] extend traditional color histograms by a statistical model that allows the extraction of both color and spatial information. This model is based on finite multiple-Bernoulli mixtures. A further example is automatic image segmentation. In this scope, Millet et al. [MBHM10] present an algorithm that combines two color-based methods working with internet databases of images. The first method separates object and background pixels, and the second one segments the image by detecting the central object. The intersection of both data sets results in the final segmentation.

Annotation of Music and Speech. Due to the improvements for storage, networking, and internet services, personal and online available music collections have grown rapidly in the recent years [TTP07]. This also results in negative effects , so that a proper storage and labeling, as well as an improvement of search and retrieval methods is required. Common music recommendation systems such as *Last.fm* [Las] and *Pandora* [Pan] exploit tags (descriptive free-text annotations, see Section 2.1.2) for query and navigation purposes [NTTM09]. Since recommendation is not possible before new tracks have been manually tagged (what is known as the cold-start problem), automatic tag annotation is applied based on audio content analysis. For instance, Ness et al. [NTTM09] describe stacked generalization as a new approach in the scope of automatic annotation. Bischoff et al. [BFP09] realize music theme annotation by exploiting existing user tags and track lyrics. A further relevant topic is *Automatic Speech Recognition* (ASR). While traditional language processing only generates words and does not consider structural aspects, Hillard [Hil08] derives an approach for automatic sentence structure detection and annotation. Wu et al. [WLD$^+$10] present a weakly supervised learning method in order to detect spoken language in domain-specific dialogue environments. For that purpose, a topic classifier is applied, and slot classifiers are trained to extract slot-value pairs. A comparison of current ASR approaches for speech segmentation, so-called regression methods, is conducted by Mporas et al. [MGF10]. As a result of employing several speech segmentation engines based on *Hidden Markov Models*, they propose a vector regression scheme that merges different boundary predictions.

Annotation of Video-based Media. As in multimedia research in general, graph-based learning methods are recently followed in video annotation research [THW$^+$09]. These methods are applied in order to ease video retrieval and improve efficiency compared to traditional manual annotation [WHH$^+$09]. In this context, Wang et al. [WHH+09] introduce an approach regarded as multigraph-based semi-supervised learning. The scope is to tackle problems that result from training data insufficiency or the curse of dimensionality. Here, different factors such as multiple modalities, multiple distance functions, and temporal consistency are managed by learning with multiple graphs. Tang et al. [THW+09] reveal limitations of graph-based learning methods with respect to the specific characteristics of video-based media, especially in the scope of semantic annotation in multilabel setting. The correlation of semantic concepts and their interaction as well is not supported. Accordingly, a new method called correlative linear neighborhood propagation is presented that adapts semantic correlation into graph-based semi-supervised learning in order to improve the performance of annotation. Often applied by video analysis professionals (cf. Section 2.1.1), *Face Recognition*, *Shot Detection*, and *Activity Recognition* are further relevant fields of investigation. In the case of face recognition, traditional generative models such as *Gaussian Mixture Models* or *Hidden Markov Models* make strong assumptions on the independent local observation of face images. Heusch and Marcel [HM10] point out that this independency is not sufficiently applicable to faces. As a result, they propose a sophisticated model that uses a static *Bayesian Network* in order to generate relationships between striking face features. Onur et al. [KGU10] present a video segmentation algorithm based on fuzzy color histogram-based shot-boundary detection that is applied for copy detection. In the process, cuts and gradual transitions (fade, dissolve) in videos are automatically computed. Human activity recognition is a young discipline assigned to computer vision. Qian et al. [QMXW10] present a pipeline in which (i) humans are detected by non-parameter background subtraction, (ii) features are extracted from the generated human bounding boxes, and (iii) activities are recognized through a Support Vector Machine multi-class classifier. Additionally, a clustering algorithm is applied.

Annotation of 3D-Models. Analogous to annotation research for the particular media formats described above, also 3D annotation research projects currently focus automatic approaches in order to provide sophisticated retrieving. For instance, Attene et al. [ARSF09] present the *ShapeAnnotator* system that supports automatic segmentation of surface meshes and the attachment of annotation according to a given ontology. Here, also ontology properties and relations that are applied to 3D segments are processed in an automated way through topology and geometry analysis. Sin et al. support collaborators in virtual 3D spaces, e.g., technical designers. There, personal intentions and comments can be shared by means of sketches that are connected to 3D surfaces of different forms [SCL09]. In the course of automated human face recognition based on 3D-data, Sukno et al. [SGF10] present an approach that exploits projective geometry across different viewpoints of the same picture. An active shape model is generated of frontal view images that are applied to segments of images taken from different viewpoints. They tackle the problem that traditional image-based face recognition systems mostly presume frontal views of the analyzed scene.

Summary. As mentioned above, an insight into the recent main topics of digital annotation research shows that a majority of the localized work follows up the development of new or sophisticated automatic approaches. In that scope, the common goals are to improve content retrieval, and to solve the problem of time exposure as well. These findings correspond to the assumptions on which the motivation and problem definition of this thesis are built (see Section 1.1). Instead of applying *concrete approaches* of automatic annotation, the annotation process is focused as an *operational entity*, considering the tasks that have to be accomplished, the order in that potentially approaches and algorithms (in form of tools or services) are executed, and the data that is passed between these tools and services as well. Beyond that, also collaborative annotation processes have been investigated. In the following, the assistance of collaboration by annotation techniques is explained.

2.1.5 Assisted Levels of Collaboration

As already mentioned in Section 2.1.1, annotations improve collaborative practices, since they enable communities to share and transmit ideas and knowledge concerned to a relevant subject, enhancing the interaction between collaborators and computers [ABDF07, AF08b, ZBM06]. Halasz [Hal88] points out that annotation is a key activity in any collaborative endeavor. Regarded as community memory, the set of annotation forms a record of the work and communication of a specific community [CDTT04, FBT$^+$03, Mar98]. Even weak forms of collaboration are supported by annotation sharing due to the offered facilities to view other group member's results, without the obligation to actively react [AFFT04]. In particular, the interaction between collaborators is established by shared or public annotations. Various levels of collaboration are supported by different types of annotation, regarded as activity and as a product of this activity as well. Collaboration levels can be assigned to the main characteristics of collaboration: interaction, sharing, and access [AFFT04]. Accordingly, this section brings these levels into focus by elucidating the role of annotation in *group communication*, *co-authoring* (as collaborative writing), and *collaborative information interpretation* (as collaborative reading).

Group Communication

When working together, communication is an essential activity [NCS09]. Shared annotations, considered as acts of dialog (see Section 2.1.3), can establish a common forum for group of persons that frequently need to reference to shared documents [NCS09, DH95]. Examples for such kinds of working groups are teachers and students, field service workers, editors and publishers, or standards organizations [DH95]. Especially in the case of asynchronous communicational situations, an amount of research work proves the advantages of digital annotation [CGG00, Fin05, ZBM06]. Also direct synchronous settings can be supported [NCS09]. For instance, Perry reports on scenarios in which scientists discuss and work on one document in real-time [Per02]. It has been detected that digital annotations have a higher potential than traditional paper notices, and also other than digital communication media, such as news groups or mailing lists [CDTT04, DH95]. These benefits arise from a common context that is established for a group discussion and, in doing so, a context in which annotated statements have been included [Bru02]. Thus, users are enabled to recognize that context. In addition to that, discussions with respect to certain topics can be found more easily [CDTT04]. Finally, in contrast to other communication forums, annotations are able to present a collaborative discourse in a better-structured form [CDTT04].

Co-Authoring

Another level of collaboration, called collaborative authoring, requires a fine-grained exchange of information among all participating authors [CGG00]. According to Baecker et al. [BNPM93] and Neuwirth et al. [NKCM90], annotations serve as an fundamental technique within design processes performed by a working group. This especially applies to groups whose members are spatially separated. Several research projects report on the application of digital annotation for co-authoring purposes, from collaborative writing of documents such as academic publications or business reports to the group-based creation of multimedia contents. Niranatlamphong et al. [NCS09] describe that annotations are used as means of providing feedback by members the involved parties such as writers, consultants, editors, and reviewers. The same principles are applied by professional VLSI (very-large-scale integration) designers, who annotated their design objects in order to state remarks, but also to provide background information [CK91]. Finke developed a system for collaborative video linking whose resulting artifacts are interactive audiovisual documents that can be used for example in the scope of e-commerce or further education of technical skilled employees [Fin05, RFHB09].

The different annotation types concerning annotation-aided group authoring are investigated by Zheng et al. [ZBM06]. Based on the analysis of the email exchange between scientific groups, general types of artifacts are identified: (i) to-do items, (ii) summaries of new entries, (iii) discussions that often include parts of the copied text, and (iv) comments-on-comments. Hence, relevant annotation forms are concluded. Whereas work-lists and edits are strongly related to the authoring process, several forms of communicational contributions are identified, such as comment, meta-comment, and reply.

Collaborative Interpretation

Collaborative interpretation of documents is defined as a process in which a group of users interprets and modifies subsets of information units in order to create coherent sets of descriptions [CG00]. Pea and Hoffert describe [PH07] such procedures as decomposition and recomposition of information. The main objective is subsumed as sense-making, i.e., information parts that by themselves provide less value must be interpreted and revised to become useful data. Cox and Greenberg [CG00] assume that the best ways of supporting collaborative interpretation through annotation are based on the manipulation of elements in a spatial medium that enables

- organization of visual items and iterative establishing of relationships,
- informal information restructuring that allows an easy addition of emergent after-thoughts, and
- easy freehand sketching in order to explore alternatives, reduce information, and create alter items [EM97].

A further relevant topic of collaboration in the context this research work has been the field of Computer-Supported Collaborative Learning (CSCL), particularly in the phase of performing the empirical study. In the following section, the roles of annotation in eLearning are elucidated, also in the sense of collaborative learning scenarios.

2.1.6 Multimedia Annotations in eLearning

Regarded as a possibility of highlighting relevant parts of a text or adding keywords and comments to the sidebar of a book, annotation is considered as the most applied learning activity, especially important in traditional learning that is based on reading activities [BR07, HWS07]. According to Marshall [Mar97], the act of annotation improves reading as well as the understanding of a text. Even when appearing as small entities, annotations can be helpful means of learning. For instance, learners may only need keywords to initiate advanced thinking processes of remembering facts that were in mind when creating the annotation [BR07, HHD08]. Respective studies have shown that the access of annotations is to be seen as more relevant than the activity of writing down an annotation [BR07]. In that context, respective key features of annotation systems are the supply of annotation repositories that, for instance, in collaborative settings, enable learner groups to share this additional information, and also to search for annotated data in different (external) sources [BR07]. Consequently, annotations help learners to be aware of the specific conventions, parameters and requirements of a larger community in which learning takes place [Rob08].

Several theories and assumptions referring to the benefits of annotation-aided learning can be outlined. Ovsiannikov et al. [OAM99] assess three aspects of learning (or crucial cognitive functions [BLR03]) that are supported by annotation: *memorization*, *thinking*, and *clarification*. Remembering is aided by the possibilities of recollecting the main points of the given contents. Comments, critical notes, or questions that reflect own opinions may be annotated attending running thinking processes. Finally, annotators are enabled to personify information by transforming it to own (verbal) representations and, in doing so, supporting clarification that is phrased in their own conceptual language. In addition to these aspects, Robert [Rob08] classifies three types of learning that can be supported by the application of annotation techniques: (i) *knowledge* in the sense of obtaining cognitive and mental skills, (ii) *attitude* with respect to the increase of feelings or emotions at an affective level, and (iii) *psychomotor*, i.e., manual or physical skills. Jonassen et al. [JDC^{+}95] emphasize the role of annotation as a constructive cognitive operation, constituting essential elements of learning theories of constructivism. In constructivism, annotations are learning mechanisms that afford third-party commentary, information manipulation in the sense of sorting or filtering, or labeling through metadata and semantics [Ahe05].

Dual Code Theory

In the scope of annotation of digital multimedia, the *Dual Code Theory* by Paivio [Pai86] is often referenced to as the argumentation basis related to the application of hypermedia in learning settings. This theory assumes that two delimited cognitive coding systems are involved in reception processes: a *non-verbal* and a *verbal* system.

- *Non-Verbal.* Responsible for the representation and transformation of non-verbal objects that describe pictorial, i.e., figurative and visual-spatial information. This information is stored within so-called *imagens*.
- *Verbal.* Processes linguistic information. This kind of information is gathered through reading or hearing and is stored in so-called *logogens*.

Paivio points out that the verbal and non-verbal systems are activated by the sense channels of the human sensory system. Accordingly, reading a certain word activates the verbal reception system. In a secondary step, also the non-verbal system can be initiated, if the same word is imagined in a pictorial manner. Consequently, a picture of a certain item will activate the non-verbal system and afterwards the respective linguistic term may be invoked. Paivio states that the consecutive activation of both coding systems results in an improved memorization capacity. The reason is that, although both systems work independently, there is still a connection that can be activated as well. In doing so, the largest probability of memorization is given when information is coded dually, i.e., verbal and non-verbal, at the same time. Hence, a dual coding of information leads to an improvement of learning effects and achievements.

A Cognitive Model of Multimedia Learning

The research work conducted by Mayer [May01] deals with the specification of design principles for multimedia learning environments that assist knowledge acquisition. He analyzed forms of processing multimedia contents by human memory and developed

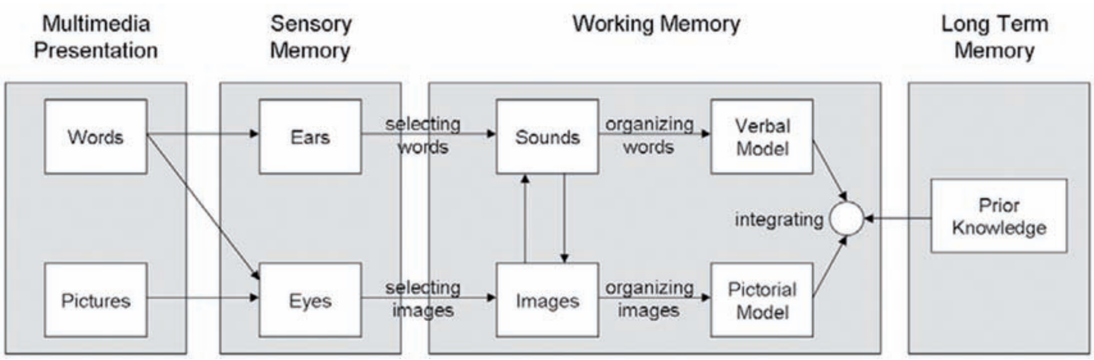


Figure 2.12: Cognitive Model of Multimedia Learning: Cognitive processing of two different information channels. [May01]

a model named *Cognitive Model of Multimedia Learning*, focusing on the question of how multimedia has to be adapted in order to enhance human learning (see Figure 2.12). Fundamental quintessence of the model is that the design of multimedia learning environments particularly needs to support the cognitive processes that are involved during learning activities. This approach is described as Learner-Centered-Design and refers to the findings of Norman [Nor93] described below.

Mayer's model is based on three basic suppositions that characterize multimedia knowledge acquisition:

1. Human cognition comprises two channels that both handle with auditory and visual information (cf. [Pai86]).
2. The ability of assimilating information through both channels at the same time is delimited (cf. [SC94]).
3. Information assimilitation through the auditory and visual channels is an active cognitive process that aids the construction of coherent mental representations (cf. [Wit89]).

Mayer identifies three general cognitive processes that apply when learning takes place: *selection*, *organization*, and *integration*. *Selection* means that the relevant auditory and/or visual information is transfered from the sensory memory to the short-term (working) memory. *Organization* occurs when coherent verbal and pictorial representations from both memories are formed in the working memory. Finally, through *integration*, knowledge is constructed by assigning verbal and pictorial representations within the working memory to previous knowledge that is stored in the long-term memory.

According to these aspects, Sweller and Chandler distinguish between two types of cognitive load. If the given content contains a lot of information units that are related to each other in complex structures, an *intrinsic cognitive load* is existent. On the other hand, extraneous cognitive load refers to the editing and presentation of multimedia contents. For instance, when the material is not sufficiently well-structured in its presentation, learners need to supply higher cognitive effort to comprehend deeper interrelations in order to effectively deal with the information. In the case of a transformation of cognitive load to a *cognitive overload*, learning can be decreased or even retarded [SC94]

Individual Learner Types and Learning Styles

The consideration of different types of individual learners is a relevant aspect in educational research. Previous investigations have shown the benefits of annotations in the scope of differing learners and learning preferences. Hwang et al. [HWS07] conducted a study about the annotation of multimedia web contents. Based on Witkins categorization of independent and dependent learner cognitive styles [WDF+62], they present a web-based annotation tool that supports self-assistance by allowing learners to create own annotations on the Web. As a result, they indicate a correlation between cognitive learning styles and learning effects and success [HWS07].

In this context, Norman [Nor93] defines varying cognitive modes, learning phases, and learning styles that should be supported by learning environments. He distinguishes between two cognitive modes that require different forms of technology support: An experimental mode refers to the perception and reaction to events, and encourages the acquisition of facts and skills; furthermore, a reflective mode is consistent with the comparison and contrast of information as well as decision making, what assists restructuring activities. Besides, different learning phases can be identified: (i) conceptualization of the learning item and its topics, (ii) the construction of new knowledge by relating the learning subject to existing mental representations, and (iii) a phase in which learners expresses own understandings and share this with other learners or teachers in a dialogue act. Finally, Norman localizes multiple learning styles and preferences that determine the best forms of learning for different individuals. In this scope, he refers to the visual, auditory, and kinesthetic perceptual styles. Further styles are also presented, such as reflector, pragmatist, theorist, or activist. As stated above, Norman indicates that an effective learning environment provides a flexible support of different learning styles that are preferred by individual learners, and fosters the acquisition of skills that refer to not-preferred styles [Nor93]. According to Normans work, Chambel et al. [CZF04] affirm the benefits of annotation-based learning systems in order to support different cognitive modes, phases, and learning styles.

Annotation-aided Collaborative Learning

Hwang et al. [HWS07] recapitulate scientific findings about the benefits of collaborative learning with respect to successful learning and teaching activities. Inter alia, they point out that collaborative learning fosters the formation of critical opinions. Here, an important factor is the ability to perform collaborative communication and discussion. That helps learners to clarify ideas and to organize and structure the common learning process. Thus, meta-cognition of students is encouraged. Furthermore, more implicit learning effects can be achieved, such as teamwork skills. They also employ an annotation system that supports previously listed aspects by implementing respective learning models, such as learning together, student team learning, or group investigation.

In Section 2.1.5, the benefits of annotations for different levels of collaboration - communication, interpretation, and co-authoring - was elucidated. These benefits can also be mapped to collaborative learning. With respect to group communication, Finke [Fin05] describes a global collaborative learning scenario in which participants construct common knowledge while they work spatially and temporally separated. Here, a collaborative system supports communication and information sharing by exploiting worldwide networks like the Internet. One case of collaborative interpretation is reported by Pea et al. [PLR06]. Group-based analysis of video material by means of collaborative annotation systems is described, pointing out the facilities for writing individual interpretations and reflections about the given content, and sharing this information with other group members. In this context, collaborative annotation systems are means of obtaining a common agreement. Mikova and Janik [MJ06] and Seidel et al. [SPK05] characterize this issue as inter-rater reliability. Stahl et al. [SFZ06] and Zahn et al. [ZHF$^+$05] describe co-authoring as a fundamental element of project-based-learning, reporting on university courses in which students collaboratively construct hypermedia documents by means of annotation in form of multimedia hyperlinking. In doing so, interactive multimedia presentations are created by the students in a self-contained manner. This form of teamwork-based learning is called *learning-by-design* [Fin05, SFZ06]. Further examples for the application of annotation systems in collaborative learning scenarios are presented in Section 4.1.

2.2 Workflow Management and Process Modeling

The general goal of this work is to achieve an improvement of multimedia annotation systems with respect to system operability regarded from different viewpoints. This improvement is to be obtained through the application of Workflow Management (WfM) Principles. In doing so, annotation processes are specified by means of Business Process Modeling standards. This section presents an introduction of WfM and workflow process modeling specifications. First of all, a general definition of the related terms is presented. Subsequently, Workflow Management Systems and respective Process Definition Models and Languages are specified. Finally, different forms of workflows are elucidated by illustrating various perspectives and patterns, providing examples with respect to multimedia annotation cases.

2.2.1 Terminology of Workflow Management

Relevant terms with respect to the field of Workflow Management are *Business Process*, *Workflow (Management)*, *Workflow Management System*, *Process Definition*, *Activity*, and *Workflow Participant*. This terminology is based on the specifications defined and provided by the *Workflow Management Coalition* (WfMC) [WFM]. As non-profit institution, the WfMC aims at developing a common terminology and standardization in order to enhance the exploitation of workflow technology. In doing so, common functional areas of workflow management products are identified. In this manner, interoperability between respective systems and also further services (such as e-mail and document management) is to be fostered by using common standards for multiple functions.

Business Process

Definition 1 (Business Process). *A set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships. [WFM99]*

According to this definition, the WfMC defines that a business process is assigned to task-related objectives and organizational relationships, such as engineering design processes or banking transactions [WFM99]. A business process may be specified for a single organization, but may also specify relations between different connected organizations, e.g., customer-supplier relationships.

A business process also indicates conditions that are associated with its initiation as well as with the results at its completition. Furthermore, the formal and informal interactions between all participants (see Definition 6) of the process are involved. Business processes may also comprise automated and manual activities, at which automated operations enable workflow management (see Definition 2).

Medina-Mora et al. [MMWF93] point out that business processes are implemented as material and/or information processes. A material process joins and delivers physical objects. Hence, involved human tasks are to be classified as physical. For example, these include moving or collecting physical products. Information processes comprise fully or partially automated tasks such as tasks performed by computer applications or tasks performed by persons in interaction with a computing system. Information is created, processed, managed, and provided, based on a given organizational structure and the environment of information systems. In order to establish information processes, databases, transaction processing, and distributed systems are applied. Accordingly, this thesis regards annotation processes as business processes that are instantiated as information processes.

Workflow (Management)

Definition 2 (Workflow Management). *The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules. [WFM99]*

Workflow (or Workflow Management [WFM99]) has to be regarded as a business process that is executed by means of computer applications [Sch00]. The general objective is to maximize the efficiency of a working process in order to achieve an overall business goal [Fau00, Hol95]. This process comprises various logical steps (tasks), relations between tasks, execution rules, and participants [CB04]. Single tasks can either be processed by humans, or executed by information technology in an automatic way. The automation is based on specific procedural rules and control data that are defined in form of a workflow model, usually called *Process Definition* [CB04, Fau00, WFM99]. There, tasks defined within the process are associated with participants, as well as with the concerned documents and the tools that are required to perform these tasks [Fau00]. The term workflow is often used in conjunction with *Business Re-engineering*. This comprises activities of analysis, modeling, definition, and the subsequent procedural execution of an organization's core business processes [Hol95]. Consequently, workflow management is an effective solution, since a separation of the process and the underlying technology is provided. This ensures the integration of frequent process changes into the procedural rules that are specified in the process definition [Hol95]. In this context, advantages of workflow management are the development of dynamic applications, the utilization of resources for flow-rate amplification purposes, as well as a reduction of the performance effort [SCJ$^+$05].

There are several distinctions drawn between different forms of workflow products [NLW05, WFM99]. Alsop [Als94] classifies workflows into ad hoc workflow management, automation of task execution, automation of process, and information flow. Frye [Fry94] presents three workflow categories: mail-based, document-based, and process-based. A distinction between human-oriented and system-oriented workflows is conducted by Georgakopoulos et al. [GHS95]. Furthermore, Ader [Ade97] classifies production workflows, administrative workflows, ad hoc workflows, and collaborative workflows. These distinctions serve as basis for the classification of workflow management systems illustrated in Section 2.2.2.

Workflow Management System

Definition 3 (Workflow Management System). *A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications. [WFM99]*

According to the definition cited above, a Workflow Management System (WfMS) can be regarded as a technological environment that enables the definition, execution, management, and control of workflows by means of a specific computer application. Hereby, the order of tasks, events, or activities is driven by the process [NLW05]. A detailed introduction of WfMSs is illustrated in Section 2.2.2.

Process Definition

Definition 4 (Process Definition). *The representation of a business process in a form which supports automated manipulation, such as modeling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc. [WFM99]*

As mentioned above, the (automatic) execution of processes is based on specific rules and control data that are defined in form of a model. Here, tasks to be accomplished are assigned to participants, documents, and the respective tools. The workflow is defined by model that is specified in a *Process Definition Language*. The resulting document is also regarded as *workflow schema* [LAH08, Sch00]. The general functions of a process definition are presented in Section 2.2.3.

Activity

Definition 5 (Activity). *A description of a piece of work that forms one logical step within a process. An activity may be a manual activity, which does not support computer automation, or a workflow (automated) activity. [WFM99]*

Activities are the fundamental components of a workflow schema [Sch00]. They form the smallest units of working processes that are monitored by workflow engines, but may comprise multiple work items that are assigned to a workflow participant [WFM99]. One needs to distinguish between manual and automated activities. The execution of a manual activity requires the involvement of a human processor, whereas automated activities only can be performed machine-aided. In that sense, the terms of *human* and *mechanical resources* are established.

Workflow Participant

Definition 6 (Workflow Participant). *A resource which performs the work represented by a workflow activity instance. This work is normally manifested as one or more work items assigned to the workflow participant via the worklist. [WFM99]*

According to the distinction between manual and automated activities, a workflow participant can be either a human processor or may involve mechanical resources such as virtual agents. Participants is localized by reference within the process definition. Besides of human and mechanical resources, also roles and organizational units may be declared as participant of a workflow instance [WFM99].

2.2.2 Workflow Management Systems

In order to support internal processes and to realize effective workflow management, distributed environments are required that (i) are component-oriented and allow integration and interoperability between mutually delimited modules, (ii) support workflow applications that correspond to business or information processes, and (iii) enables the further development, integration, and replacement of suchlike applications during process reengineering [GHS95].

These requirements conform to Workflow Management Systems (WfMS), which are regarded as the organization´s key systems in order to foster competitive convenience, customer support, and productivity [CB04]. In general, such systems ensure the automation of business processes by managing the sequence of tasks and the activation of human and technological resources assigned to these tasks [Hol95]. WFMSs are computer environments that enable the execution, management, and control of workflows with respect to the order of events and support the structured specification of such processes [NLW05].

The primary function is the enactment of case-driven workflows [vdAHKB03]. In this scope, administrative and monitoring functions are provided [WFM99]. WfMSs comprise different interconnected tools, applications, or software components [CB04, WFM99]. These (sub-)systems are applied in order to (i) specify, schedule, and store concrete workflow processes, defining a set of rules that allow coordination of task order, and (ii) read and interpret specified processes to control interaction between the system and participants as well as other applications [CB04, CCT$^+$04, GHS95, WFM99].

According to the latter aspect, concrete workflow instances are interpreted by one or more *workflow engines* which are able to interact with the participants of a workflow and to invoke potential external services or applications [CB04]. In addition to that, they are also responsible for coordination, sharing, and transmission of information, documents, and tasks which are passed between workflow participants during processing [CB04]. Whereas traditional WfMSs are only drafted to coordinate processes and its involved agents, advanced systems are able to exploit networks and web technologies supporting cross-organizational interactions [CCT$^+$04]. Furthermore, sophisticated features are provided, such as process repository maintenance, workflow adaption, matchmaking, exception handling, and process documentation and reflecting [CCT$^+$04].

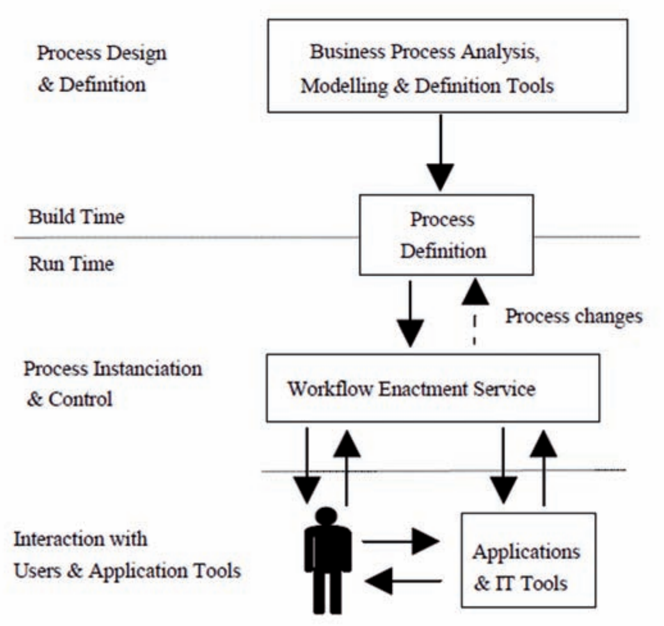


Figure 2.13: Characteristics of Workflow Systems [Hol95].

At an abstract level, Hollingsworth [Hol95] classifies WfMSs according to three fundamental functional areas: *build-time*, *run-time control*, and *run-time interaction*. Build-time functions are deal with the definition and modeling of workflow processes and the comprised tasks and activities. The area of runtime-control concerns the operational issues of the environment, managing the workflow processes and the tasks as part of each process as well. Finally, run-time interaction functionalities are provided in order to support interaction with workflow participants and tools related to different activity steps. Figure 2.13 illustrates these main functions and their relationships.

Schätzle [Sch00] summarizes differing classifications of WfMSs that can be localized in literature. These categories are based on (i) the *type of processes* that are supported (ii) the *concept* used to *model* processes, and (iii) the *basic technology* applied to realize the workflow engine.

I. Classification according to supported Process Types

Based on the categorization of different forms of workflows presented in 2.2.1, WfMSs can be distinguished according to the types of workflow processes that they support.

- *Production Workflows.* Such systems are applied in order to enact workflows that represent the key processes of an organization. The focal point is a reliable and effective work handling. These processes comprise a high rate of iterations that have equal properties among themselves and, consequently, gather a high level of structuring.

- *Administrative Workflows.* Analogous to production workflows, administrative workflows disclose a high level of structuring. In contrast, they are less complex according to the set of activities and obtain fewer importance for the organization.

- *Ad-Hoc Workflows.* WfMSs that support ad-hoc workflows need to handle low-structured processes that are executed in differing manner between the iterative steps. Besides, alternating roles and responsibilities have to be considered, as well as a stringent demand for communication.

II. Classification according to Process Modeling Concepts

The main criteria for this classification are the concepts of the workflow definition language applied by the WfMS. Four groups of systems are differentiated: (i) form-oriented, (ii) communication-oriented, (iii) conversation-oriented, and (iv) process-oriented.

- *Form-oriented Systems.* Form-oriented WfMSs regard documents as key elements of a process description. The documents are enriched with additional routing information that indicates which is the receiving participant after each workflow step. From a technical point of view, form-oriented systems are realized by means of Email. As such, they are suitable for more simple processes. Complex processes that include multiple documents and concurrent processing can hardly be handled by these systems. One reason is that the process description is not managed by a central unit, but allocated at different documents.

- *Communication-oriented Systems.* These systems are realized based on the supposition that process coordination requires communication. In this scope, coordination is defined, for instance, by modeling so-called communication graphs. Similar to form-oriented systems, these WfMSs are also mostly implemented as Email systems.

- *Conversation-oriented Systems.* Starting from the assumption that humans act through speech, the *speech act theory* is the basic idea of conversation-oriented systems. Varying kinds of speech acts can be identified that constitute a process as network together with participants. From this point of view, conversation-oriented systems can be regarded as a specific form of a communication-oriented system.

- *Process-oriented Systems.* Here, the process as such is in focus, and the information about the process to be coordinated is defined by a coordination entity. Thus, a logical and central coordination is enabled. The process is regarded as a sequence of activities which are associated with incoming and outcoming documents or data. In doing so, complex processes can be mapped.

III. Classification according to applied Technologies

A classification that is traced to the applied technologies distinguishes between email-oriented and database-oriented systems.

- *Email-oriented Systems*. WfMSs that are based on Email exploit mail technology in order to present and deliver documents and task descriptions to participating persons. Suchlike mail systems obtain advanced functionalities such as rule-based forwarding or electronic signature. Besides, there are further specialized workflow applications that have been augmented with respect to routing purposes and Email functionality.

- *Database-oriented Systems.* Typically, a (Relational) Database Management System forms the basis that stores workflow data as well as routing and status information. The workflow engine is mostly provided by a central server application. Client applications are only responsible for information representation. Although the installation and enactment of such systems may be a laborious work, they are outstanding because of their high performance and operational reliability.

2.2.3 Process Modeling

This section deals with the concepts for modeling of workflow instances. First, the terms *Workflow Model* or *Workflow Schema* as well as *Process Definition Language* are classified. Next, a categorization of Process Definition Languages is illustrated and relevant examples of languages are presented.

Workflow Schemes and Languages

As stated above, the automatic execution of a process by a WfMS is driven by a workflow model which represents the specification of an ideal workflow process and is defined before the process starts [Fau00, LAH08]. In this manner, workflow instances can be defined at the task, as well as the at structure level [GKK$^+$08]. Workflow models are instantiated according to specific cases, for instance, a tax declaration or an insurance claims [vdAHKB03]. Accordingly, the required processes and logical steps have to be modeled and embedded into the system in the form of a schema or a specification. *Workflow schemes* are applied in order to define a process by means of a *Process Definition Language* which serves as instrument for documentation, analysis or execution by a WfMS [Sch00]. A workflow schema is composed of a network of activities, relations between activities, process start and termination conditions, and further activity-related information such as workflow participants, connected applications, data, etc. Furthermore, it may reference separately defined sub-processes which are also elements of the overall process definition [WFM99]. The Process Definition Language provides modeling constructs. Accordingly, the expressive excellence of a language determines explicitly the number of processes that can be supported by the system. The descriptive elements of a definition language enable mapping of the control flow (see Section 2.2.4), i.e., the activities and sequential relationships can be expressed. In this manner, causal independencies between activities can be described in order to achieve a sequential execution, and concurrent activities can be configured. In addition to that, a Process Definition Language facilitates the specification of the data flow (see Section 2.2.4) in which particularly incoming and outgoing data, but also human and mechanical resources, are assigned to each single activity. In general, there are two types of workflows that can be specified within a workflow schema: *abstract* and *concrete workflows* [GKK+08] (see also the description of the BPEL language below). Abstract workflows describe activities in an abstract manner, not referring to the specific related resources. Concrete workflow models assign the specific resources to the executable tasks, providing information about service semantics and execution.

Classification of Process Definition Languages

Kiepuszewski [Kie03] introduces a classification of Process Definition Languages with respect to, among other, the consideration of different *Workflow Perspectives* (see Section 2.2.4) and *Workflow Patterns* (see Section 2.2.5). Here, the following four classes of workflow models are presented [Kie03]:

1. *Standard Workflow Models.* Standard workflow models are related to Process Definition Languages that consider basic control flow patterns and multiple workflow instances. There, loops and process termination are not bounded to specific restrictions. Furthermore, workflow states are not defined.

2. *Safe Workflow Models.* As opposed to standard workflow models, multiple instances are not supported, i.e., it is not possible to obtain different instances of the same activity at the same point in time.

3. *Structured Workflow Models.* As a subclass of standard workflow models, structured models refer to languages in which so-called AND-Splits or OR-Splits (see Section 2.2.5) are always followed by a respective AND-Join or OR-Join. In addition to that, arbitrary loops are not supported, i.e., loops have explicitly one entry and one exit point.

4. *Synchronizing Workflow Models.* Synchronizing workflow models bound boolean values to each single activity. If a true token is sent, the receiving activity will be executed. In the case of a false token, the activity is passed on. Synchronization points expect boolean values from any incoming branch and propagate respective true or false tokens.

Examples of Process Definition Languages

Up to now, several different languages have been developed with various objectives and striven application domains. This section presents an excerption of selected Process Definition Languages. First, *BPEL* and *XPDL* are illustrated as two mainly deployed modeling standards. Second, so-called graphical modeling languages are exemplified by *BPMN* and *YAWL*.

BPEL In order to specify business process behavior based on Web Services and business interaction protocols, the *Business Process Execution Language for Web Services* (BPEL4WS or BPEL) has become the de facto standard which is broadly adopted [CDT06, SHJ$^+$08, VSS$^+$07]. First of all, BPEL describes the orchestration of services according to the order and sequence of service execution [SHJ$^+$08]. In general, BPEL determines a set of primitive activities in order to invoke services. These primitive activities are combined into more complex activities [CDT06]. Hence, BPEL comprises the capabilities of common Process Definition Languages, supporting a two-layer modeling concept [VSS$^+$07]. The *function layer* contains web services as executable components that are responsible for the execution of basic activities. The execution order of these activities is specified in the *choreography layer*. Two types of business processes can be defined using BPEL [CDT06]. The first, an abstract process, specifies the protocol role of a business as well as its public aspects. The second one, an executable process, specifies state and logic of the process by considering the sequence of web service interactions. The pursued service-oriented approach of BPEL leads to a clear separation of the implementation level of activities and the execution environment [VSS$^+$07]. On the other side, due to the service-orientation, the flow of data can only be defined indirectly and, since the data transfer always need to pass the BPEL engine, a performance bottleneck evolves [SHJ$^+$08]. In addition to that, inter-organizational collaboration is not explicitly supported [CDT06].

XPDL The *XML Process Definition Language* is a XML-based file format adopted by the WfMC and used to transfer process models between different applications. In addition to BPEL, XPDL is another widely-used standard in the field of process modeling. A detailed description can be found at the XPDL Specification by the WfMC [WMC]. The main part of an XPDL process definition is the specification of a set of workflow processes that are to be regarded as self-contained component within a working process. Defined processes may be integrated into one superior process at which a hierarchical model is achieved. Hence, a workflow process is mapped as a directed graph where nodes are assigned to activities (tasks) and the edges determine the workflow control. A node can depict different types of activities such as manual or automatic. Manual activities are performed by human participants. In doing so, users must notify the system of the termination and results of such activities. Since not every task can be processed by each user, XPDL enables to associate nodes to one

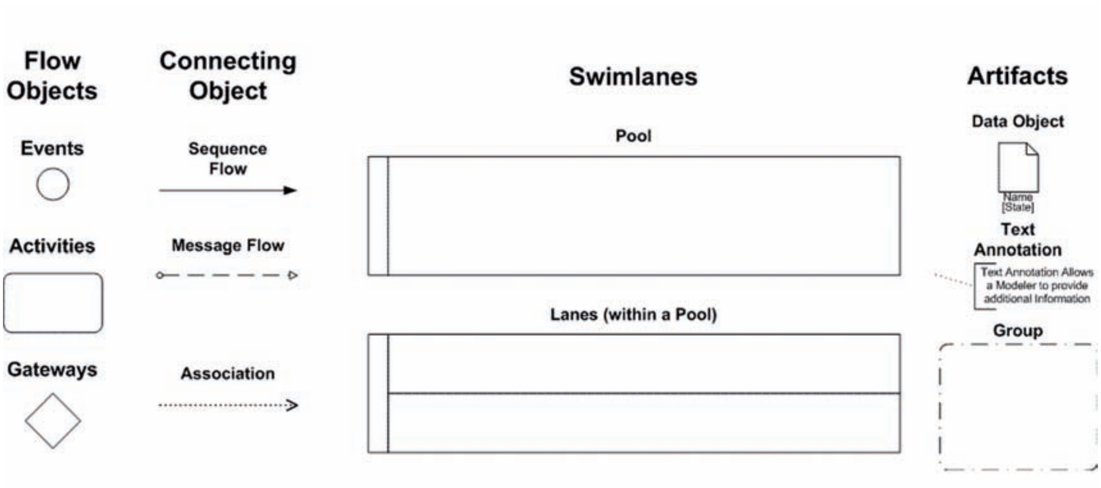


Figure 2.14: BPMN Core Graphical Elements. [BPMI]

user (or user group) that is extracted from a global user list. A concrete user management must be provided by the WfMS. Automatic activities are executed in an invisible manner for human participants. For that purpose, an XPDL file can either contain the source code of a program(-segment) that is to be executed or the interface description for an external application. XPDL is frequently used as file format for *BPMN* (Business Process Modeling Notation) which addresses the same objectives from a different perspective. BPMN is regarded as one of the *graphical* process modeling techniques. Within complex processes, graphical notations facilitate the communication between business users and technical users. In addition to that, users are enabled to interpretate the process in sophisticated manner [SPB05]. There are various graphical process definition languages such as BPMN or YAWL (Yet Another Workflow Language) which are exemplified in the following.

BPMN The main objective of the *Business Process Modeling Notation* (BPMN) that is developed by the Business Process Management Initiative (BPMI) is to provide defined graphical elements in order to represent business processes in a flow chart [Whi03]. For elucidation, Figure 2.14 illustrates the BPMN core graphical elements. By means of BPMN, workflow management and modeling by technical and business users is supported, enabling human workflow modelers to work with graphical representatives of fundamental workflow elements and hence, in addition to the software level, realize the interoperation of business processes at the human level. BPMN can be formally mapped to Process Definition Languages such as BPEL or XPDL [Whi03]. Thus, a standard visualization mechanism is provided for processes that are specified with a respective language. BPMI and WfMC started a cooperation in 2002, at which the WfMC accepted BPMN as a graphical notation for XPDL in order to improve modeling tasks. From a structural point of view, BPMN and XPDL are similar, since both are flow-chart structures. Figure 2.15 illustrates an excerption of general mappings from BPMN to XPDL (Workflow, Transition, Task, Decision, AND-Join).

BPMN Graphical Object	*Mapping to XPDL*
The details of a Pool or an Expanded Sub-Process	<WorkflowProcess/>
	<Transition/>
Task	<Activity> <Implementation> <Tool/> <Performer/> </Implementation> </Activities>
Decision	<Activity> <Route/> <TransitionRestriction> <Split Type="XOR"/> </TransitionRestriction> </Activities> Combined with a: <Transition> <Condition/> <Transition>
C D F Join AND-Join	<Activity> <Implementation/> <TransitionRestriction> <Join Type="AND"/> </TransitionRestriction> </Activities>

Figure 2.15: Mapping BPMN to XPDL. [Whi03]

YAWL Similar to the main objectives of BPMN, YAWL (*Yet Another Workflow Language*) provides a graphical notation for workflow elements [SPB05, Fou]. YAWL is an open source framework developed by the YAWL Foundation [LAH08, Fou] as a result of cooperations between Eindhoven University of Technology and Queensland University of Technology [vdAH03]. It is based on Petri-nets, overcoming the disadvantages that result from their inability to map the entire set of required Workflow Patterns (see Section 2.2.5). With respect to Workflow Perspectives (see Section 2.2.4), it explicitly supports control flow, data flow, and the operational perspective [SPB05]. Analogous to BPMN, YAWL can be translated to other Process Definition Languages. Figure 2.16 shows the symbols used by YAWL.

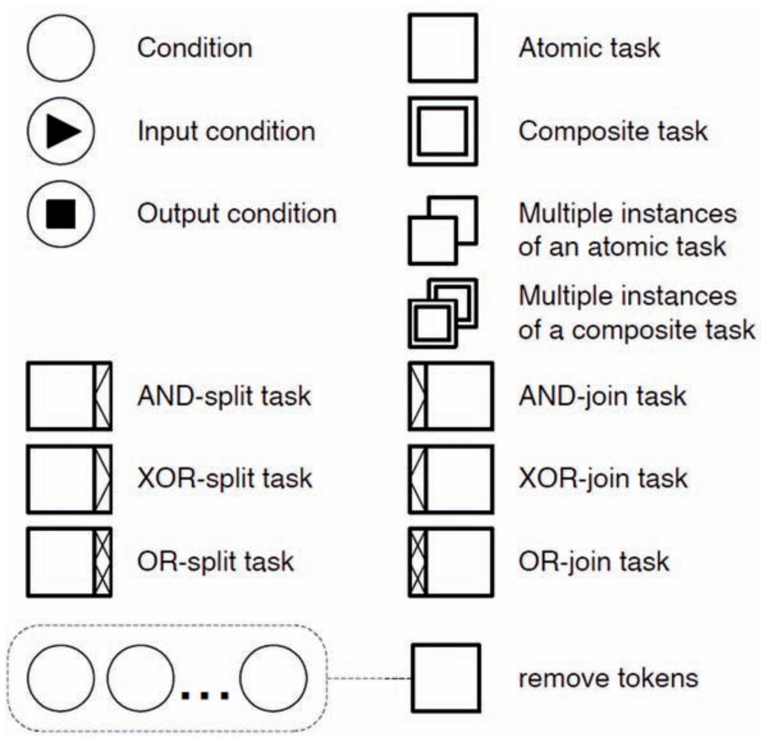


Figure 2.16: Graphical Constructs of YAWL. [vdAH03]

2.2.4 Workflow Perspectives

Workflow processes can be viewed from various different perspectives that each emphasize different integral parts of a workflow [vdAH03, vdAHKB03]. Van der Aalst and van Hee [vdAvH02] distinguish between the four general perspectives *control-flow*, *data-flow*, *resource*, and *operational*:

- The *control-flow* specifies the activities or tasks that have to be accomplished, as well as their sequential order. This is achieved by the specification of different constructors that allow control and monitoring of the flow execution, for instance, linear sequence or parallel execution. Here, single activities are regarded as atomic work units, so-called work items.

- The *data-flow* integrates the business and processing data into the control-flow perspective. These business and processing data include documents and other entities that are transfered between activities and participants, as well as local workflow variables which determine the pre- and post-conditions associated to certain activities.

- The *resource perspective* concerns the human and technological participants of a workflow instance. Here, an organizational structure anchor is provided for the workflow which defines human and device roles that execute the process.

- The *operational perspective* defines the granulated steps that are performed between activities, at which the steps map into underlying applications. By means of specific interfaces that define the connection between activities and applications, references to workflow data are passed as incoming and outgoing information. Hence, a manipulation of data within applications is possible.

2.2.5 Workflow Patterns

In the course of workflow process analysis, especially with respect to the specific requirements for Process Definition Languages, different so-called *Workflow Patterns* have been identified and determined. They describe the modular operational elements of a workflow and, in form of a pool of patterns, define the process behaviors. Workflow patterns describe processes in an imperative style expression, independently from specific Process Definition Languages [vdAHKB03].

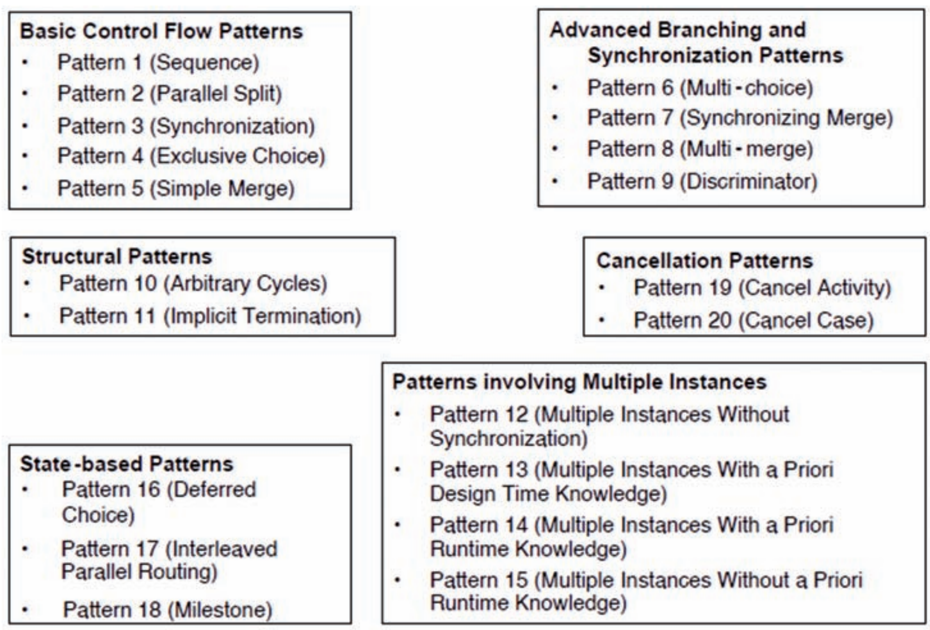


Figure 2.17: The twenty most relevant workflow patterns. [vdAH03]

Van der Aalst and Hofstede exemplify the twenty most relevant workflow patterns [vdAH03]. These patterns are categorized into six classes (see Figure 2.17):

1. *Basic Control Flow Patterns* are regarded as the basic constructs of a flow of activities. They enable the modeling of sequential, parallel, and conditional execution.

2. *Advanced Branching and Synchronization Patterns* go beyond the basic patterns and provide sophisticated types of splitting and joining flows.

3. *Structural Patterns* allow a more freely structuring of activity flows, compared to rigid block structures provided by programming languages.

4. *Patterns involving Multiple Instances* allow the multiple instantiation of workflow parts at the same time.

5. *State-Based Patterns* focus on the states of activities and further events. In doing so, essential behaviors and process stages such as milestones can be realized.

6. *Cancellation Patterns* implement occurrences that require an interruption of the running workflow, or even the cancellation of the whole case.

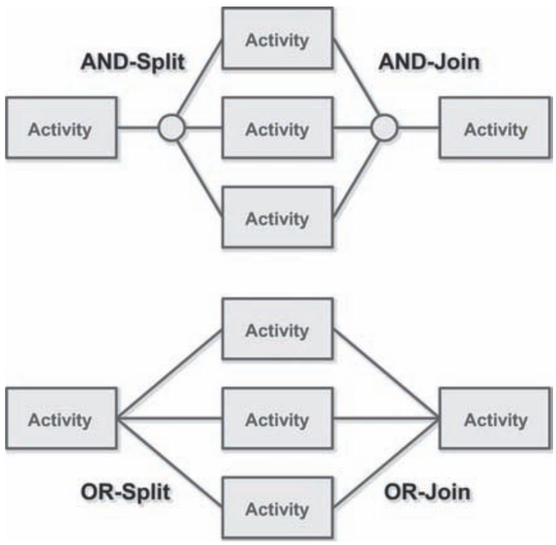


Figure 2.18: Excerption of Basic Control Flow Patterns (cf. [WFM99]).

For the purpose of illustration, the basic control flow patterns are presented in the following, exemplifying concrete situations within multimedia annotation processes.

- *Sequence.* Sequential or serial routing is given when an activity within a workflow process is subsequently invoked or enabled after the termination of the previous activity. For instance, the activity "categorize image cutouts" is executed after the completion of "execute image segmentation".

- *Parallel Split.* Also considered as AND-Split or parallel routing (see Figure 2.18), this behavior divides a single thread into multiple threads of control that can be executed in parallel manner. Hence, suchlike parallelized activities can be processed simultaneously. For example, this is the case when "detect shots" and "detect faces" are triggered after selecting a video file.

- *Synchronization.* In the scope of synchronization, previously parallely executed sub-processes are brought together into a single thread. The precondition for this is that all incoming branches have been executed only once (for the other case, further patterns are applied). A synonym term for synchronization is AND-Join (see Figure 2.18). AND-Joins are required when detected faces need to be assigned to detected video shots .

- *Exclusive Choice.* This is a point in the process where depending on a condition or decision, one of several alternative threads is followed. One example for such kind of conditional routing is when, according to the media format of a document selected for further processing, multiple tools can be invoked that each are applicable to a delimited number of (or only one) media type(s).

- *Simple Merge.* In a simple merge, two or more alternative branches are brought together that have not previously been executed parallely, so that the joining activity is performed asynchronously. For the case of parallel execution, the Multi-merge and Discriminator patterns are defined. Asynchronous merging is allowed if, for instance, segmentation approaches have been applied to two media objects of different formats such as video scene detection and image segmentation. In this case, an equivalent following activity (such as segment categorization) does not expect a synchronous continuation, since a merging of the metadata generated for each media object is not required.

The illustrated examples of multimedia annotation sub-processes show that Workflow Patterns are to be regarded as one relevant aspect with respect to visualization and execution of annotation processes. This is also the case for Workflow Perspectives, since a separated consideration of the component invocation and the transfered data is required. Thus, Workflow Perspectives (as well as Workflow Patterns) are employed in this thesis (see Chapter 6). In fact, a large amount of pattern-related research has been done, resulting in the determination of various differing Workflow Patterns that are applied to varying business process cases [vdAH03]. A detailed elucidation of the main Workflow Patterns is provided by van der Aalst et al. [vdAHKB03].

2.3 Summary and Conclusions

In this chapter, fundamentals of the main topics of this thesis - Digital Multimedia Annotation and Workflow Management - have been introduced. First, a general overview has been provided that refers to the meaning of the annotation term, from a general viewpoint of personal additional information to more global purposes of information sharing. In that context, it has been indicated that annotation not only has to be regarded as (digital) entity, but may also describe the process of adding new information. Furthermore, the transition from paper-based annotation to computer-aided information addition has been exemplified, elucidating the different functions, application purposes, and participating user roles of annotation systems.

The succeeding sections pointed to the diversity of recent annotation systems with respect to multiple application areas as well as different forms and appearances of annotations that are established by means of current technical solutions. There, the role of annotations employed in the Web, Digital Libraries, Database Management, Information Retrieval, Media Content Analysis, and Augmented Reality has been emphasized. Additionally, a classification of annotation forms has been presented that distinguishes between annotations as metadata, content, hyperspace elements, and dialog acts. As a result, these sections confirm the assumptions which substantiate the described problem definition and challenge of this work. It is clarified that, due to the versatility of modern annotation systems that support multiple types of annotation formats and functionalities, the clarity of user interfaces is decimated, which has a negative impact on learnability and usability of such systems.

Next, the influence of multimedia formats on annotation of digital data has been demonstrated. Initially, a simplified division of media formats into the two general classes continuous and discrete media has been presented. In that scope, three different definitions of multimedia(-systems) have been introduced that each emphasize the reciprocity of continuous and discrete media, the human-system interaction aspect, as well as the synchronization of different interrelated media objects. These definitions served as basis for the explanation of two relevant aspects of multimedia annotation. First, anchoring is described as the process of defining specific areas upon the media objects that determine the validity of additional information with respect to time and space they reference. Synchronization means the interpretation of relations between media objects and annotations (which can be also media objects themselves) that are defined by specified validity areas. Particularly, it has been pointed out that anchoring and synchronization are important aspects in this thesis, since structural and organizational information is determined that is transfered between subsequent tasks (and tools/services) within an annotation workflow.

The last two sections of the first part explore the role of annotations in collaborative working settings and eLearning scenarios. In doing so, it is presumed that collaborative annotation-aided learning has been a significant application area for the conducted research work, with respect to the empirical study and the implementation of the concepts as well. In this context, different levels of collaboration - communication, collaborative interpretation (reading), and co-authoring (writing) - have been pointed out, illustrating the way in which annotations support each of these activity fields. In the scope of computer-supported learning, theories and assumptions referring to the benefits of annotations in general have been distinguished, such as memorization, clarification, or acquisition of physical skills. Furthermore, the presentation of two relevant theories - Paivio's Dual Code Theory and Mayer's Cognitive Model of Multimedia Learning - elucidated how (annotated) multimedia objects support learning, considering different cognitive systems of human reception that are activated by differing forms of media. Additionally, it is shown that annotation systems are able to attend different types of learners with respect to various learning styles. Finally, in the scope of collaborative learning settings, the fostering of learning-by-design or project-based learning is emphasized in the sense of constructive learning activities. To summarize, these two sections reveal that collaborative learning can doubtlessly be regarded as one relevant application area of digital multimedia annotation. Hence, a foundation of this research work on that specific application domain is justified and revealed.

The second part of this chapter aimed at constituting a uniform terminology for the terms used in this thesis which are primarily related to the area of Workflow Management. Definitions of the significant items Business Process, Workflow and Workflow Management (WfM), Workflow Management System (WfMS), Process Definition, Activity, and Workflow Participant have been presented.

In the further course of this chapter, WfMSs and Process Definition standards have been introduced at a theoretical level, presenting the fundamental characteristics, relevant examples, as well as general classifications related to these both domains. The presentation of WfMSs serves as basis for the analysis of requirements that addresses the general design of process-driven systems derived in Section 6.1. Additionally, these requirements are associated to standards of process definition approaches which, in turn, also form the groundwork for a formal specification of annotation processes manifested in Section 6.2.

According to the latter point, it has also been illustrated that workflows might be viewed from different perspectives that have an impact on the specification of a process. These include the workflow control and the operational perspective with respect to the execution of processes and more granulated sub-activities, the data that is transfered between different workflow steps and tools, as well as the resource perspective that includes the human and technological participants of a workflow process. Finally, different basic Workflow Patterns have been introduced that determine the behavior of a process within granulated sub-steps. It has been pointed out that such patterns have to be considered within the conceptional results presented in this thesis, especially in the scope of the visualization and (semi-)automated execution of annotation processes.

While this chapter, especially the first part, provided a comprehensive and general overview for the domain of digital multimedia annotation, the next chapter will deepen this general understanding by exemplifying different classes of features and functionalities that are provided by common (multi-)media annotation systems.

Chapter 3

Feature-oriented Analysis of Multimedia Annotation Systems

This chapter presents an analysis of multimedia annotation systems with respect to provided features. The general aim is to derive incorporated tools, functionalities, or approaches that can be assigned to tasks and activities performed by users of annotation systems. Second, types of data are acquired that originate as a result of system features, and can be defined as input and output data for phases of an entire multi-step annotation process.

The underlying methodology for the identification of system features is described in Section 3.1, and is followed by the exemplification of the examined applications and their detected features in Section 3.2. In Section 3.3, the incorporated systems are compared by contrasting and subsuming the essential functionalities according to different feature classes. Finally, conclusions that particularly regard the scope of this thesis are derived in the last part of this chapter.

3.1 Methodological Approach for the Identification of System Features

This section elucidates the methodology applied to obtain specific annotation-related features which are compared and subsumed in Section 3.3.

In order to choose the applications considered in the analysis, a preliminary investigation was performed which included more than eighty annotation systems. A selection of adequate tools is to be regarded as a significant factor, since this thesis addresses various forms of multimedia annotation. Thus, a particular priority was given to the consideration of different types of systems with respect to the supported media formats as well as application domain and purposes. According to these points, a general classification was developed including the following categories of annotation systems: *Open Standards*, *Digital Document Annotation*, *Web Annotation*, *Text Annotation*, *Image Annotation*, *Audio Annotation*, *Video Annotation*, *3D-Model Annotation*, *Social Annotation*, *Semantic Annotation*, *Content Analysis*, and *Audio and Video Transcription*. The entire pool of pre-examined annotation tools and the association to different classes is listed in Appendix A. With regard to the division of systems into different categories, it must be clear that classification overlaps are not taken into consideration.

In the next step, ten applications were selected for detailed investigation which represent different categories of annotation systems. These include *A.nnotate*, *ATLAS.ti*, *AnnoCryst*, *ELAN*, *EVA*, *EXMARaLDA*, *MADCOW*, *M-OntoMat-Annotizer*, *Music Annotator*, and *Vannotea*. Selection criteria were the level of distribution (for research prototypes, whether they are used at all), referencing in research papers, and the range and relevance of provided features. In order to work out features and functionalities of these systems, annotation runs were conducted using the respective software, and/or instruction manuals and online tutorials were surveyed.

After detecting all relevant features for one application, tasks or activities were derived by determining purposes and goals for the application of the respective features. In doing so, a process-related category system was inductively developed. That is, categories which represent different tasks or activities were directly derived from the material by means of generalization, at which the set of categories was modified after each examination run (cf. [BD06]).

According to different types of tasks or workflow phases, the presentation of identified features (next section) and the subsequently conducted comparison (Section 3.3) are based on the categories *System Configuration*, *Area Selection*, *Information Attachment*, *Search and Exploration*, *Import and Export*, *Process Assistance*, and *Collaboration*. The first five categories represent general phases of the annotation workflow at a sufficient level of abstraction. As central topics of this thesis, *Process Assistance* and *Collaboration* basically relate to features which can be assigned to the support of (collaborative) annotation workflows.

3.2 Multimedia Annotation Systems

In this section, the results of the extraction of annotation-related features is presented for each incorporated system. First, a brief introduction is given including the recital of supported media formats. Second, the relevant elements of the graphical user interface are illustrated. That followed, identified features are explained according to the derived process-related categories.

3.2.1 A.nnotate

A.nnotate is a web service that allows users to upload, annotate, and share digital documents such as PDF, Microsoft Word, or image copies of websites [Anna]. In its early stage, it was developed by *Textensor Limited* in the course of a project funded by the Scottish Government. The first resulting prototype *Notate* is described in [CH07], at which the main focus was semantic web authoring. Since 2008, the company distributes a standalone versions as well as a specific API that allows web application developers to enrich CMSs (Content Management System) with annotation functionalities. For instance, shelf modules are offered for *Moodle* [Moo]. According to the product website, the services are exploited by different types of customers such as researchers, students, designers, web developers, or learning organizations [Anna].

After logging in, a user can immediately upload a desired document which, for sharing purposes, is stored on a central server. The supported file formats are PDF, Microsoft Office files such as XLS, PPT, DOC, DOCX, etc., images in PNG, JPG, and GIF, as well as the open document standard ODF. Additionally, a *snapshot* tool is provided that allows user to create a copy of a web page and store it as image.

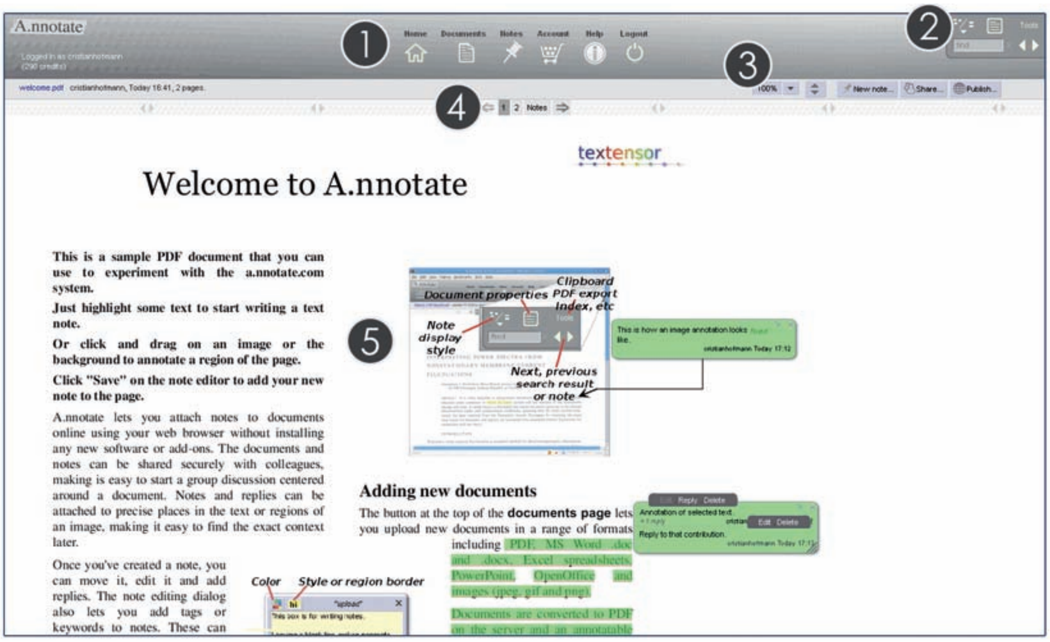


Figure 3.1: The document annotation and sharing service *A.nnotate*. [Anna]

As Figure 3.1 implies, the graphical user interface is divided into five sections. In the upper center area (1), specific icons are placed by which users can switch between different views of the system. The *Home* view gives general information about the activities a user (with its specific account) has performed. The *Document* page represents all documents that are treated by the user account and provides several functionalities for search and manipulation. Analogously, all annotations generated by the user are represented and managed in the *Notes* view. Account-specific customization is provided in the *Account* page. Top right, configuration options for the user interface, the main tools of the service, as well as a simple search form are provided (2). Beneath this is a toolbar that enables activities on the document level (3). Among these are zooming, create new annotation (for the entire document), share, and publish. (4) is a bar element that allows navigation between the pages comprised in the open document. Furthermore, the visualization of annotations can be (de-)activated. The main work area (5) represents the document and its integrated annotations, as well as the main annotation tool as pop up window.

System Configuration. The most relevant configuration options concern the visual appearance of the user interface elements. For instance, besides of defining the color, annotations can be displayed in three ways: (i) next to the specific document area an annotation refers to, (ii) on the right-hand margin in form of a list, and (iii) as footnotes at the bottom. Furthermore, several default document properties can be set such as the status of new generated annotations (private or shared), or specific policies for other users, for instance, if they are allowed to change the status of annotations, create tags, or to comment.

Area Selection. Besides of the entire document, A.nnotate also distinguishes between contained text and image areas as selectable fields for connecting annotations. Text areas can be marked either by clicking on a single word, or by dragging the cursor upon a certain passage, paragraph, etc. The visual appearance of the mark can be customized subsequently. Here, available options are *highlight* (like a highlighter pen), *insert* (text is underlined), *striketrough*, or *hyperlink* (text is underlined blue). Areas of images are selected by creating bounding boxes that enclose the area of interest (by means of mouse dragging). Alternatively, a single element can be selected by clicking. After selection of an image field, the area is highlighted, and an arrow is created that is linked to a subsequently opened dialog window. In addition to that, a website specified through the snapshot tool can be annotated directly after uploading.

Information Attachment. After selecting an area of interest, the annotation dialog is displayed. For any kind of selected content, the most relevant types of potential additional information are free-text comments and one or more tags. With respect to the tagging facilities, a user is able to either enter new keywords, or select some from a set of predefined tags. Moreover, annotations may also comprise the following information: (i) a subject or title, (ii) author (account name), (iii) creation date, (iv) related document page, (v) the status (private, shared, feedback), or (vi) a URL. The annotation dialog also provides options for subsequent editing of annotations, as well as for creating a new replying annotation.

Search and Exploration. A string-based search feature is provided. Here, users can search for documents or annotations, based on attached meta information. In the case of annotations, contained text as well as assigned tags are matched. By means of two specific buttons placed next to the respective input field, navigation between different objects resulting from a search query is enabled. A further related option provided on the main workspace is document page navigation. At the *Home* and the *Notes* view, documents and annotations belonging to the current user account can be explored. Here, the representation of objects can be customized (folders, list, thumbnails, number of displayed elements). Displayed annotations can be sorted according to properties such as creation date, tags, or related document. In addition to that, the representation can be filtered with regard to the type of annotation, for instance, selecting only annotations on text or tagged entities .

Import and Export. For printing or email forwarding purposes, documents can be exported to PDF files, including the document content and its annotations (according to the chosen annotation display form). Furthermore, annotations of one document can be saved as CSV (Comma Separated Values) file.

Collaboration. Any uploaded document can directly be shared with other service users. For this purpose, reviewers are invited via email. These users are allowed to attach their comments and tags on a read-only copy of the document. Furthermore, discussions around one document or selections of it can be conducted. By means of the *reply* button on the top of the annotation dialog, new comments can be inserted that refer to the initial text. Similar to traditional web discussion boards, replying contributions are attached beneath its parent annotation.

3.2.2 ATLAS.ti

ATLAS.ti is a commercial annotation system that is applied for the analysis of text, audio, and video documents, especially in the scope of investigational working settings. The focus lies on qualitative analysis and is based on grounded theory, and the primary purpose is to perform analytical, interpretative, and governing work. Accordingly, ATLAS.ti provides features for data management, extraction, analysis, comparison, and aggregation of gathered data. Nevertheless, it has to be regarded as an instrument for qualitative research in general. Potential application areas are psychology, criminology, text linguistics, literature, medicine, etc. The initial development started in the scope of an interdisciplinary research project named ATLAS at the Technical University of Berlin, Germany. A commercial distribution is followed up from 1993 on by *ATLAS.ti Scientific Software Development GmbH* [Atl].

ATLAS.ti supports a wide range of media and file formats. Among these are *Plain Text* (Unicode, Double Byte Character Sets, with restrictions DOC and HTML), *Rich Text* (including hyperlinks and embedded objects, e.g., XLS, PPT, images, videos), and *Images* (JPEG, BMP, and TIFF). Audio and video files can also be annotated. Here, supported formats depend on the installation of a *Windows Media Control Interface*[1] (since the application runs only on Windows systems). The latest version, *ATLAS.ti*

[1]Windows MCI is an interface to control abstract multimedia devices and resources. For more information see http://support.microsoft.com/kb/142731.

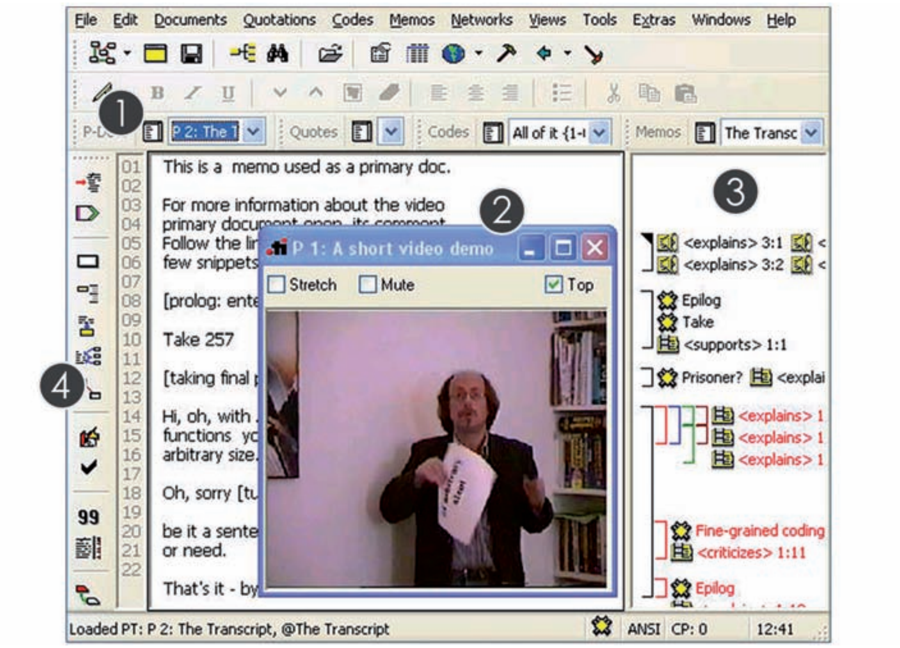


Figure 3.2: Graphical User Interface of *ATLAS.ti*. [Atl]

v6, is enabled to process PDF documents and geological data. For this purpose, a *GoogleEarth* [Goo] plug-in is integrated.

The most important elements of the main workspace are the *Object Drop-Down Lists* (1), the *Primary Document Pane* (2), the *Margin Area* (3), and the *Primary Document Toolbar* (4) (see Figure 3.2). Four drop-down lists represent the main entities of an ATLAS.ti project: documents, segments (selected areas), categories that arise from a specific vocabulary, and so-called memos (short private comments). The *Primary Document Pane* displays the selected documents to be annotated. Within the right-hand *Margin Area*, annotated categories, memos, and hyperlinks are visualized. Additionally, specific brackets indicate for the position with respect to related document segments. The *Document Toolbar* provides the relevant annotation functionalities.

System Configuration. General properties concerning system configuration and user interface appearance, e.g., size and position of windows or font size and color, can be adjusted in the *General Preferences* dialog. Additionally, user management is provided by means of a specific user database. Each user is assigned to one account by an administrator. But, no explicit management of rights and policies is realized (permission assignment is realized for each document by its initial author). Besides of user management, also data source management is provided.

Area Selection. Depending on the media format of the current document, different forms of areas of interest can be selected. Selection mostly happens manually with one exception: by means of the *Auto Coding* tool, text can be segmented into words, phrases, etc. In the other cases, selections are either created in the *Quotation Manager*, or drawn by means of mouse dragging. That is, a character, word, sentence, paragraph, or the entire file can be marked. Inside graphic documents, rectangular bounding boxes are drawn. For continuous media formats, points in time as well as time intervals can be defined. Finally, snapshots of the current state of the GoogleEarth plug-in can be defined that are treated as new media document.

Information Attachment. Multiple types of additional information can be localized. These include *Relations* that can be placed between different documents and different segments, as well as *Codes* (categorical item from a vocabulary). Further annotation types are *Memos* as short private contributions, and *Comments* as free-entered text for description, interpretation, and reflection purposes. Comments may be attached to entire documents, segments, and also relations. Here, segments can be assigned to more than one comment. Categorization of segments can be performed in different ways from the *Main Menu*, the *Vertical Bar*, the *Code Manager*, or simply open a menu by right-clicking. Usually, a segment is associated with categories by dragging them from a specific list into the marked area. In textual documents, it is also possible to define selected text as category.

Search and Exploration. In the so-called *Object Explorer*, all elements of the current project are visualized in an hierarchical view. Within the *Code Manager*, categories of the used vocabulary are represented. If a category is selected, related segments are displayed including their context within the associated media document. This list can be sorted and filtered according to specific properties. In this context, users are enabled to define own filters by phrasing new search expressions. Categories, segments, comments, and memos are also visualized in the *Margin Area*. There, the information structure can be modified through Drag&Drop interactions. Moreover, the *Family Manager* tool allows clustering and grouping of documents, categories, and memos, at which so-called *Families* are established. Multiple search functionalities are provided. Besides of traditional keyword-based or string search, also pattern matching and categorization search has been implemented. By means of the *Query Tool*, more complex searches can be performed that are based on combinations of categories. In general, search queries address media documents, segments, categories, memos, families, and relations. Lists of search query results can also be filtered.

Import and Export. Three file formats can be exported. HTML is used to generate summaries of the annotated data of a project. As one standard file format, XML can be interpreted by a large amount of applications. Thus, storing annotations in XML is supposed to foster cross-platform interoperability. Furthermore, SPSS is exported in order to continuously process annotated data by means of statistical software.

Process Assistance. Multiple media objects can be defined as “document-to-annotate” within one single ATLAS.ti project. Thus, a certain workload may be defined. In addition to that, one automatic approach is integrated that releases manual human effort. The *Auto Coding* tool finds relevant text passages, selects a specific set of text, and assigns these passages to categories that have been previously marked.

Collaboration. For information sharing purposes, ATLAS.ti can be installed at a central server application that is connected to multiple clients. Then, documents can be stored in an accessible shared directory on the common network. As mentioned above, no explicit access and rights management is provided by the system. But, the initial author of a document is able to assign specific policies. Any document and its including objects obtain information about the (currently logged-in) account name, creation date, and creation time.

3.2.3 AnnoCryst

As visualization of scientific knowledge, 3D crystallographic structures are exploited in various disciplines such as organic chemistry (biomolecular protein modeling, DNA visualization, etc.) or inorganic chemistry (design of functional nano-materials, etc.) [HHK07]. *AnnoCryst* is an open source system that allows annotation of several types of crystallographic structure formats, especially focusing collaborative work in electronic research [Cry]. It is a product of the *AA1 workpackage* at the *DART* project, a collaboration between the University of Queensland, Monash University and James Cook University, funded by the Australian Commonwealth Department of Education, Science and Training. AnnoCryst is based on the *Annotea* framework - a web-based annotation server developed in the scope of the *W3C Semantic Web initiative* [Annb] - and extends it with annotation functionalities for 3D crystallographic models [HHK07]. AnnoCryst supports the following crystallographic structure file formats: CIF, PDB, XYZ, and MOL.

AnnoCryst comprises several annotation-related tools as shown in Figure 3.3. A general toolbar (1) provides the most relevant tools of the system. An integrated *Annotea Sidebar* (2) [Anna] enables searching and browsing within the generated set of private and shared annotations. The representation and manipulation of the annotated 3D-Models is enabled by one or two *JMOL*[2] components (3). By means of the *Resources Explorer* (4), 3D-Models can be retrieved from external resources. Not shown in Figure 3.3 are the *Annotation Dialog* tool for annotation generation, the *Annotation Graph*, a graph-based visualization tool for annotations and connected replies, and the *Semantic Search* tool that enables tag-based queries on external resources.

System Configuration. Taking up the annotation concept provided by Annotea, the system enables metadata association by means of different vocabularies and annotation schemes. As the most striking configuration-related feature, vocabularies can be imported, created, and edited within the AnnoCryst environment.

Area Selection. Selection of areas-to-reference is based on the graphical structural components of a 3D-Model, for example, a represented molecule. For this purpose, the respective element has to be clicked.

[2]JMOL is an open source viewer for crystallographic models. For more information see http://jmol.sourceforge.net

Figure 3.3: Crystallographic 3D-Model Annotation with *AnnoCryst*. [Cry]

Information Attachment. Once a graphical sub-element has been selected, the *Annotation Dialog* tool can be invoked, which enables entering of annotation metadata and body content. Annotation metadata refer to its specific type such as *Comment*, *SeeAlso*, *Question*, *Explanation*, *Advice*, *Semantic Annotation*, etc. Additionally, policies can be defined that determine access rights in the context of collaborative work. The annotation body can be represented by different forms of information. First, this may be a free-text, by which also discourse is possible if the annotation refers to an already existing annotation. Second, hyperlinks to external web resources can be entered as an URI, as well as a path of a locally stored file (image, audio, video, pdf, or a further crystallographic model). Furthermore, semantic annotation is enabled through the association of semantic information that can be gathered from an available ontology. Finally, tagging is allowed by means of a selected scheme, vocabulary, or ontology. Here, annotations can be connected to academic resources and websites provided by the tag-based web services *Connotea*[3] and *Delicious* (see Section 2.1.2).

Search and Exploration. In the initial phase of annotation, the *Resources Explorer* is used in order to search and retrieve 3D-Models from online databases (e.g., ICSW or PDB) or local institutional repositories (Fedora or DSpace). The loaded 3D-Model can be viewed in the integrated JMOL viewer, at which the representation can be manipulated by rotate, pan, and zoom interactions. Annotated areas are color-marked. In

[3] Connotea is a web reference management service for clinicians and scientists. For more information see http://www.connotea.org.

the so-called *two-model layout*, two different views of the same object are obtained. With respect to annotations, search and navigation features are provided by the *Annotea Sidebar*. Here, a hierarchical visualization of annotations is presented. Furthermore, annotation metadata as well as body content is showed for a selected annotation. Within the *body* panel, potentially contained multimedia files are also displayed. The annotation visualization provides filtering facilities based on specific attributes. In addition to that, the sidebar allows string-based search that is based on annotation metadata and body information as well. Based on free categorization pre-conducted through tagging, the *Semantic Search Dialog* allows SPARQL queries on external resources such as the user's work station or academic resources. As an alternative to the *Annotea Sidebar*, sets of specific annotation types can be viewed by additional tools. For instance, tags can be presented by a so-called *Tag Cloud*. Furthermore, in the scope of team discourse, the *Annotation Graph* represents all relations between linked annotations.

Collaboration. Asynchronous communication is enabled by the possibility of annotate on existing annotations (both textual). Thus, a discussion around the given 3D-Models is established. For this purpose, other authors are explicitly invited. The configuration of respective policies determines the access on "first author" annotations. This also applies to annotation attachment in general, which can be conducted by several team members in both synchronous and asynchronous collaborative work. In doing so, the features provided by the connected *Annotea* application are exploited. In the context of asynchronous collaboration, *RSS Feeds* are used in order to send notifications to authorized authors when changes have occurred on the shared data set.

3.2.4 ELAN

Developed at the Max Planck Institute for Psycholinguistics of Nijmegen, The Netherlands, *ELAN* (EUDICO Linguistic Annotator) is a linguistic annotation tool that was designed in order to provide a sound technological basis for creation, editing, visualization and retrieval of audio and video recordings [SW08, WBR$^+$06]. In its initial phase, the support of speech and gesture research was emphasized. Up to now, ELAN has evolved to a system for the annotation, analysis, and documentation of multimedia data sets, for example in the domains of language, gesture, or sign language. The latter case has been enhanced in the EU project *ECHO* (European Cultural Heritage Online), in which data from different signed languages were collected, annotated, and published [BR04]. Depending on the underlying media framework installed on the actual operating system (DirectX, Quicktime, or JMF), ELAN supports different video and audio file formats.

As illustrated in Figure 3.4, the workspace consists of five main panels. The *Video Viewer* (1) is able to display up to four video files at the same time, for instance, to present different views of the same scene. To the right, a specific tab navigator

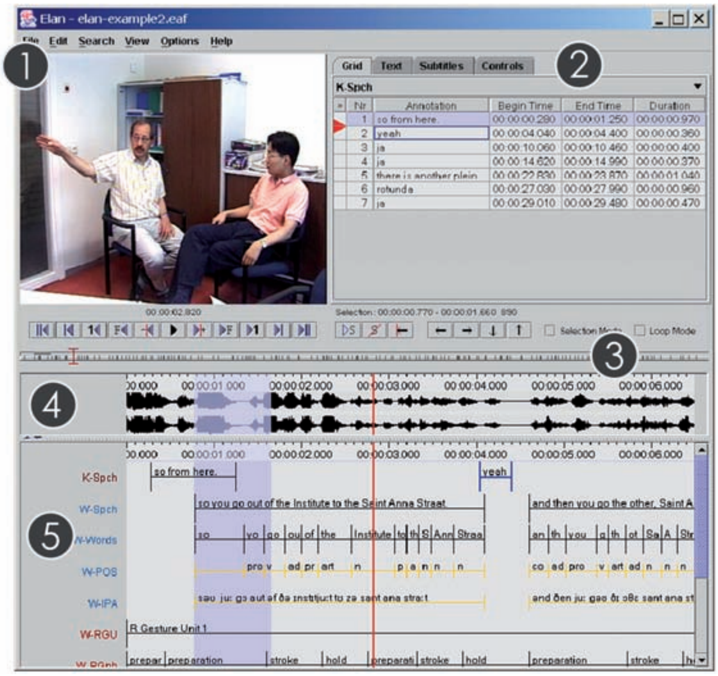


Figure 3.4: *ELAN* (EUDICO Linguistic Annotation). [LAT]

panel (2) shows meta information as well as the content of given annotations on the one hand, and on the other hand control elements for the recently active video viewer are provided. A timeline-based component (3) presents the frequency of annotations with respect to the playback-time. The waveform diagram of a (separate) audio track is visualized in the *Waveform Viewer* (4). The waveform is synchronized with the *Timeline Viewer* (5). Here, different layers of annotations and their values are showed with regard to their appearance in the playback-time.

System Configuration. The layout for the user interface and its elements is widely customizable. For instance, this applies to the color coding of annotations, the size and position of panels and viewers, the selection of displayed viewers, or the resolution for time-dependent visualizations. With respect to the used annotation scheme and methods explained below (which are also customizable), the annotation language can be determined. Moreover, the system allows import and export of preferences as templates.

Area Selection. In general, ELAN works on the basis of time intervals. As these can be specified on the level of milliseconds, points in time are supported implicitly. Time-based selection may be performed in different ways and from different panels. On the *Timeline* and *Waveform* viewers, areas are defined through Drag&Drop. Alternatively, the start and end time of an interval is specified by keystroke. On the *Video Viewer*, it is possible to click on the display area, and the boundaries of selections can be modified subsequently. When generating a new selection, three alternative modes can be chosen which determine whether existing selections are potentially overwritten or shifted. In addition to that, also annotations can be selected and referenced.

Information Attachment. ELAN realizes a specific structural annotation concept. First, tiers can be defined that are represented as layers in the *Timeline Viewer* and can be regarded as category of annotations. Accordingly, tiers comprise sets of annotations that share the same properties. In this context, two types of tiers are provided: *independent tiers* that comprise annotations connected to a selected time interval and *referring tiers* that contain annotations associated to annotations of other tiers. Attached annotations are free-text or terms according to a predefined vocabulary. Information can be attached in different ways, for example by clicking on the timeline, by means of a specific edit dialog, or using fixed key shortcuts. A further type of annotations are links that can be set between multiple media files. For instance, the same situation recorded by different cameras can be interconnected.

Search and Exploration. Multiple access points are provided for content and additional information. Time intervals can be selected on the *Go-To* window or by shifting a red *Crosshair Line*. Furthermore, users are allowed to “jump” step-by-step through a document, for instance, switching between the time borders of a selection, or from annotation to annotation. Moreover, different views of annotated data are provided by the top right view, enabling navigation and editing of tiers, time selections, and annotations. This panel additionally comprises multiple control elements in order to playback and navigate through active media files. Several search alternatives are implemented. A text search component permits queries on the recent project, at which a *search history* as well as *query storage* and *export of search results* is provided. In addition to that, more detailed and structured search through multiple projects is featured by the *substring*, *single layer*, and *multiple layer* tools.

Import and Export. The interoperability with familiar applications is specially focused, so that a large list of file formats of annotation data can be imported and exported as well. Among these are *Shoebox/Toolbox* files, *Transcriber* files, *Chat* files, *Praat* files, *Tab-delimited Text* files, *TigerXML* files, or traditional transcript files (cf. Appendix A).

Process Assistance. First of all, multiple media files can be opened at the same time, so that completeness of annotation is fostered. Furthermore, activities of area selection and information addition can be conducted by means of keyboard shortcuts. Selection can be automatized by creating series of successive selections that have the same length. Additional automatization is provided by the *Audio Recognizer* tool which analyzes the audio track with respect to silent passages. Recognized silence is automatically defined as selection area and annotated with the value “silence” by default. Further selection support is given by the possibility of creating a new area at the end point of an existing selection by means of double-clicking. Moreover, explorative processes are fostered by several forms a step-by-step navigation.

Collaboration. Collaboration on a simple level is facilitated through merging transcription files that derive from different resources. More comprehensive, ELAN can be connected to Peer-to-Peer networks, allowing a user (acting as administrator or chair) to create an *annotation session*. Here, participating users need to register via email account or an assigned id. In the process, the administrator makes the required media files available which can be either downloaded or streamed by the other session members. In that scope, the session chair is allowed to assign sets of annotation layers to different users. Then users work on their local copies and publish the results to their co-annotators which, in turn, can accept or deny changes. The focus is on synchronous work. Accordingly, a chat tool assists to real time communication, at which viewers can propose changes before committing. For the same purposes, one user at the same time is allowed to point on the respective area, for instance, selecting an existing time interval or an annotation, or setting a point or area mark upon the video display.

3.2.5 EVA

EVA (Efficient Video Annotation System) has been developed at the IBM Research Intelligent Information Management Department [IBM]. It is a server-based web application that fosters distributed collaborative annotation of semantic concepts in large collections of images and pre-segmented videos [VSN05]. A main focus of design was the improvement of usability aspects as a consequence of a conducted analysis on the former product *VideoAnnEx*, which provided a set of automatic approaches but failed with regard to operational aspects. In general, EVA aims at simplifying and accelerating the annotation process without increasing organizational effort, reducing errors such as incompleteness of annotation, and enabling sufficient configuration options in order to support different user preferences and annotation styles [VSN05]. In addition to that, collaborative work is supported by facilitating distribution of workload and inter-annotator analysis. EVA supports video and image file formats depending on the installed media framework.

On the main area of the user interface (see Figure 3.5 (1)), a customizable number of thumbnails is provided that represent either images or video shots. The thumbnails are presented in several pages, which can be accessed page by page through specific navigation elements (2). Left-hand, a list of available categories is provided (3). Each category can be assigned to one of four labels that are provided by respective buttons (5). A progress bar (4) indicates for the work state with respect to a preselected amount of media objects that have to be annotated.

System Configuration. Besides general user account management, EVA comprises several customization options with regard to annotation and manipulation of data representation. Among others, the number and size of displayed thumbnails per page, as well as their organization in columns may be configured. Furthermore, the vocabulary used during annotation and a set of media files can be defined.

Figure 3.5: The IBM *EVA* Annotation System. [IBM]

Area Selection. With respect to video content, pre-segmented files are processed that have been computed into video shots and stored in MPEG-7 format [4]. Hence, no explicit selection is scheduled, since either entire images or video keyframes (representing a shot) are annotated. Instead, more than one file can be selected at the same time to be equally annotated.

Information Attachment. The basic additional information are terms that derive from a small controlled vocabulary of thirty-nine words. These terms are in accordance with visual concepts (rather than temporal or aural) that are contained by the selected images or frames. In addition to that, an annotated concept can be marked up with one of four labels at each object, giving the annotation additional assessment: (i) *positive*: file contains the concept definitely, (ii) *negative*: file does not contain the concept, (iii) *ignore*: semantics of file is vague, and (iv) *skip*: file remains currently not marked up. Besides of traditional manual annotation, also keystrokes can be defined in order to support labeling.

Search and Exploration. The navigation-related features are limited to page-by-page and thumbnail-by-thumbnail navigation by means of a virtual cursor. In the case of collaborative task distribution, a personalized view is provided that accords to the assigned workload, that is, user obtain access only to enabled sets of data. Search functionalities could not be identified.

[4] In the use case reported by Volkmer et al. [VSN05], shot detection was performed by the Fraunhofer Heinrich Hertz Institute in Berlin, and MPEG-7 encoding at the Dublin City University

Process Assistance. EVA users are not only allowed to operate the application by means of mouse. The process also can be entirely controlled by keyboard. In the course of annotation activities, a virtual cursor can be used in order to navigate between thumbnails. Labels can be assigned only using one keystroke. Additionally, once a label has been attached by keyboard, the cursor automatically advances to the next thumbnail. Depending on a specific previously assigned workload, a progress bar visualizes the annotation progress (in percent) for a given concept. Moreover, the entire progress is illustrated in the start page. In the so-called *bulk-mode*, entire sets of images and shots, a whole page, multiple pages, etc., can be labeled equally at the same time.

Collaboration. As the most striking feature with regard to collaboration, users with administrative rights are allowed to distribute tasks or workloads (sets of data) to individual users. Furthermore, they may also shift work between different users during the running process. Hence, the collaborative annotation processes can be regulated. Since EVA is a server-based system, the system and its comprised data can be accessed through a web browser. In doing so, the physical distribution of large collections of content and annotations are resided on server-sided databases.

3.2.6 EXMARaLDA

EXMARaLDA (Extensible Markup Language for Discourse Annotation) integrates and manages different standardized file formats, concepts, and tools for the transcription and annotation discourse, as well as for the generation, analysis, and evaluation of spoken language content [Exm]. EXMARaLDA is being implemented at the Collaborative Research Center on Multilingualism at the University of Hamburg, Germany [Sch09]. The primary goal is to develop a common framework that enables sharing, exchange, reuse, and archiving of heterogeneous multilingual data between different projects of the center. However, the single products are made freely available for non-members of the research center. A provided data model is time-based and at the same time resembles common data models used by systems such as *AGTK*, *Praat*, *ELAN*, *TASX*, or *ANVIL* (see Appendix A) in order to foster cross-platform interoperability. With the same objective, XML is used as storage file format, and characters are encoded in Unicode. The framework is able to process audio and video file formats that are either supported by the JMF, DirectShow, or Quicktime.

The most important integrated EXMARaLDA tools are the *Partitur Editor* (adding transcriptions in musical notation, synchronization between transcription files and digitalized media files, segmentation of transcriptions into linguistic segments, see Figure 3.6), the *Corpus Manager* (pooling transcriptions into common entities, addition and retrieval of metadata on transcription corpora), and *EXAKT* (search and retrieval within a corpus, contextualized views of query results). In the following, the operational work with the main component *Partitur Editor* will be illustrated.

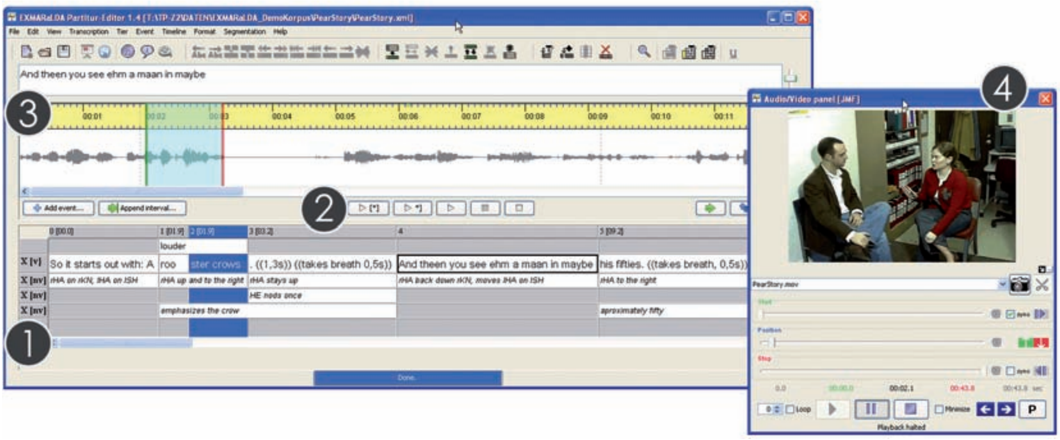


Figure 3.6: The *EXMARaLDA* Partitur Editor and Audio/Video Panel. [Exm]

Within the EXMARaLDA workspace, the *Partitur* window (1) represents different layers which contain transcriptions (annotations) assigned to time intervals. In the middle area (2), controls for the opened record as well as selection and annotation tools are provided. The so-called *Oscillogram View* (3) presents a timeline and a waveform visualization of an audio track. It is only active if the transcription is connected to an audio or video record. Otherwise, the waveform diagram is not displayed. Outside from the *Partitur Editor*, the *Audio/Video Panel* (4) comprises additional features for representation and control of media files.

System Configuration. In the *Preferences Dialog*, user-defined configurations can be performed that refer to the visual layout and presentation of the environment or properties for the annotation process. Among these are font appearance, loading of style sheets, preferred segmentation algorithm, annotation language, used media players and frameworks, presentation of annotation for printing purposes, or tools that are displayed or hidden on the user interface. With respect to the user interface layout and presentation, size, position, and color can be customized for respective elements. In addition to that, vocabularies or sets of categories can be created and edited, which are presented in the *Annotation Panel* during the process.

Area Selection. In general, time-based spaces can be selected as areas-to-annotate on the *Partitur*, *Timeline*, and *Oscillogram* views. The latter provides different options for selection creation and editing such as click and drag for initial creation, click and mouse wheel for length (duration) modification, or shift selection boundaries through keyboard shortcuts. Time areas may also be merged and splitted. Additionally, selection boundaries can be manually entered as time-based data within both the *Navigation* and *Event* dialog boxes. The *Partitur View* allows setting of time-based marks that are assigned to a point in the audio or video playback-time. In this context, either an image of a video frame can be created, or a specific segment of the audio track is extracted by means of the *Snippet* tool. Realtime selection, that is, defining start and end times during playback, is enabled by using the "space" key. Here, also a fixed maximal length of time intervals can previously be defined. EXMARaLDA supports the integration of automatic segmentation algorithms that are applied on the annotation timeline. But the results are not presented in the user interface, they are only used in the context of other applications.

Information Attachment. The general annotation concept includes events that are related to time-based selections within the playback-time of audio or video content. These selections can be assigned to images, audio files or videos by inserting an URL or path referring to the local file system. Events are represented in the *Partitur View*. These entities are synchronized with the audio or video track. Furthermore, a selection can be categorized by terms that derive from one or more specific schemes which have to be regarded as systems of hierarchically interlaced categories. Categories are assigned by selection on the respective item provided by the *Annotation Panel*. Events are transcribed by entering free-text. In addition to that, special characters according to the *International Phonetic Alphabet* can be inserted. On a higher level that refers to the entire document, general information can be attached such as *project name*, *transcription name* (which serves as title of exported HTML and RTF files), *convention*, *referenced file*, or *comment*. Moreover, layers on the Partitur may be described by the properties (i) *speaker*, (ii) *type* (transcription, description, annotations, link), (iii) *category* (free classification similar to tagging), and (iv) *display* (layer name). Finally, points of interest can be stored by defining bookmarks on the *Timeline* or *Oscillogram View*.

Search and Exploration. The waveform of an audio or video track can be explored on the *Oscillogram View* by means of pan and zoom. Here, a step-by-step navigation through the existing time selections is enabled. Navigation and control of the track is also featured by the *Audio/Video Panel*, at which a specific cursor is synchronously shifted on the *Partitur View*. For visual clearness purposes, layers on the *Partitur View* can be sorted and filtered according to specific properties. EXMARaLDA provides text-based search in events, i.e., in any annotated free-text. In the process, the property *search area* determines which layers have to be examined. A sequential navigation through search results is permitted by the *find next* functionality that is applied on the resulting list. In addition to that features, a specific integrated search tool, *EXAKT*, provides more detailed querying possibilities.

Import and Export. In general, file formats delivered by different familiar applications can be imported and interpreted. Among these tools are *TASX*, *Annotation Graph*, *ELAN*, *FOLKER*, and *Praat* (see Appendix A). Accordingly, respective annotation file formats also can be exported. With regard to file export, all annotated data is managed and stored in a XML format, so that data uniformity is guaranteed. In addition to that, EXMARaLDA supports export of RTF, HTML, TXT, and combined HTML and SWF (HTML and Flash Player for media display). Finally, results from search queries can be exported as text file. Information in any generated formats can be printed out.

Process Assistance. Features that provide rudimentary process support have been identified. First, keyboard shortcuts can be exploited in order to work with time intervals. That applies to selection, annotation, and navigation activities as well. Second, step-by-step navigation through time selections is provided in the *Oscillogram View*, what may assist to the information addition process.

3.2.7 MADCOW

An application that is frequently referenced in relevant research literature is *MADCOW*, developed at the University of Rome "La Sapienza", Italy from 2002 on. In its first version, MADCOW has been applied in the field of Digital Libries [AF08b]. Later, especially the commercial *MADCOW Lite* version has been developed for the annotation of Web documents [BCL+04, BLL+06]. In general, MADCOW supports the generation, manipulation, interlinking, and retrieval of multimedia annotations on digital documents such as websites (in the DL-context also text, image, video, etc.). In addition to that, collaboration is supported in order to establish a network of decision-based documents. For this purpose, a client-server architecture is provided at which HTTP-servers are responsible for a centralized storage of annotations. Annotated documents are any kind of web pages containing multimedia including text, images, audio, and video objects (but just text and images at the *Lite* version).

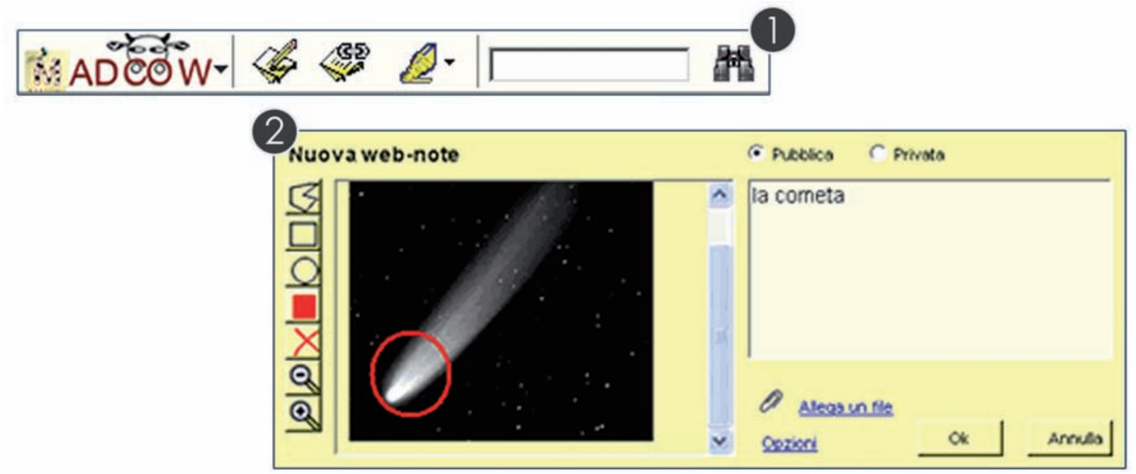


Figure 3.7: The *MADCOW* Toolbox and Image Annotation Dialog. [Mad]

The MADCOW user interface is located as plug-in toolbar within a web browser[5] (see Figure 3.7 (1)). As such, the toolbar consists of a menu comprising configuration and personalization functions, as well as four function buttons that feature annotation generation, insertion of a link to an external resource, marker-like highlighting of annotations, and - in conjunction with a respective input field - search on the document.

[5] *MADCOW* is applicable to Microsoft Internet Explorer and Mozilla Firefox.

Area Selection. In general, content provided within a document is selected by simple user interactions such as click and drag to mark a text passage, or clicking of an embedded object. Embedded media objects can optionally be farther segmented. For this purpose, media-specific tools are invoked after selecting an embedded object. The images tool (see Figure 3.7 (2)) provides features to superimpose lines or shapes (circle, rectangle, polygon) in order to highlight the part of interest. A video-specific tool (not supported by the *Lite* version) enables users to determine start and endpoints of time intervals referring to the video playback-time.

Information Attachment. A MADCOW annotation contains two main components. First, metadata is assigned such as the annotation type, author, creation or modification date, location, URL, or public/private visibility. With regard to the type of an annotation, the different values *explanation*, *comment*, *question*, *solution*, and *summary* are pre-defined. The second component of an annotation is its content. This is a HTML coded document and is not only restricted to text (written by the user), but may also contain links to other multimedia objects such as text, images, audio files, or videos. These multimedia documents can also be annotated in turn.

Search and Exploration. Once a document is loaded in the web browser, placeholders that indicate for the location of annotations are displayed upon the document. Therefore, the system searches for existing annotations within connected servers. Different icons stand for different types of annotations. When selecting an icon, the annotation object is opened in a separate web browser window. For presentation filtering purposes, the menu on the MADCOW toolbar offers options in order to show or hide annotations, or to show only own annotations. A textual search is realized through an input field and a respective button. in the process, also public annotations are examined that have been integrated by other users. Query results are listed in a new web browser window.

Collaboration. MADCOW is implemented as client-server-architecture. While the client is a plug-in for standard web browser applications, the server(s) can be regarded as repositories of annotations to which clients can log-in. All annotations stored in the servers are public, i.e., they build the basis for a shared collection of information. As already described, media objects that serve as content within an annotation can in turn be annotated. In doing so, a rudimentary form of discussion around web documents may be established.

3.2.8 M-OntoMat-Annotizer

Developed inside the *aceMedia* project, *M-Ontomat-Annotizer* is a system that represents a combination of the original *OntoMat-Annotizer* application and a specific plug-in named *VDE* (Visual Description Extractor) [Mon]. As a reference implementation of the *CREAM* framework [HSM01], the basic system Ontomat-Annotizer supports generation and maintainance of ontology-based OWL-annotations on textual documents. M-Ontomat-Annotizer (as VDE plug-in) extends the former application by allowing annotation on image and video files [BPS+05]. Here, the main focus lies on the aggregation of MPEG-7 visual features extracted from the media files with semantic concepts provided by a (RDF(S)) domain ontology [PAS+06]. In doing so, visual descriptors can be expressed in a machine-understandable format [ATP+05]. Besides of Text and HTML, M-OntoMat-Annotizer is able to process video (MPEG, MPG, AVI, MOV) as well as image files (GIF, JPG, JPEG, TIFF, TIF, PNG).

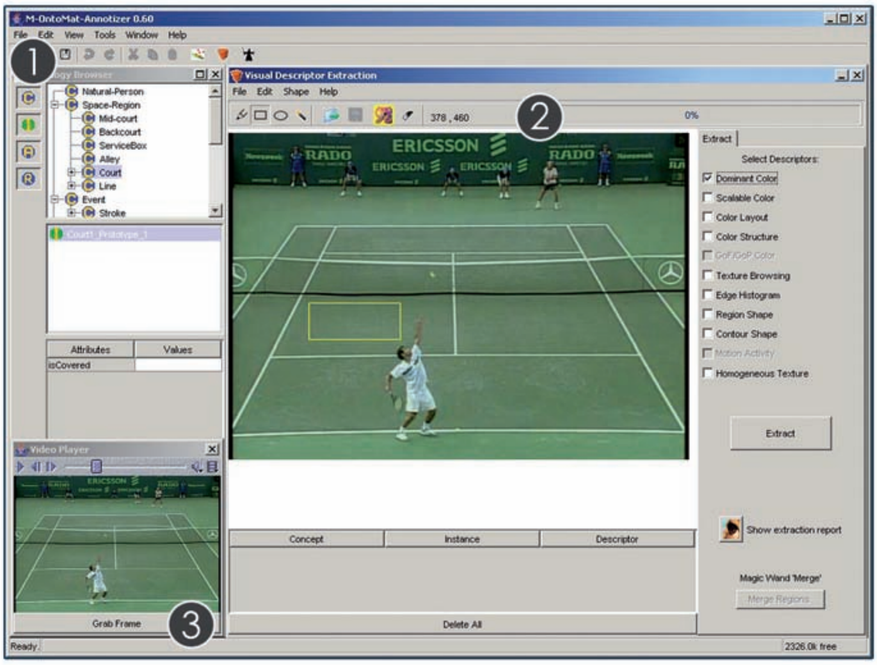


Figure 3.8: Semantic multimedia annotation with *M-Ontomat-Annotizer*. [Mon]

When the *VDE* plugin is active, the M-OntoMat-Annotizer user interface can be divided into three relevant components (see Figure 3.8). The *Ontology Browser* (1) is a standard component of OntoMat-Annotizer and is responsible for the visualization of a loaded ontology in hierarchical form, including the ontology's structure, instances, and semantic relations. Images or representative frames of videos are displayed in the *Visual Description Extractor Panel* (2). Here, also potential selections of parts of the media files are visualized, and generated annotations are listed. Furthermore, visual descriptors that have to be extracted are provided. On the bottom left side, the *Video Player* (3) is displayed if a video file has been opened. Here, control elements are provided, and a video frame can be selected through the *Grab Frame* button in order to be displayed as image in the *VDE* panel (1).

System Configuration. The *Ontomat Options* dialog allows customization of general options such as the used host and port in order to communicate with a connected server application. Moreover, a directory can be chosen for the storage of ontologies that are accessed by the *Ontology Browser*. Optionally, an assistant tool can be activated. By means of the *Manage Tools* dialog, different integrable plug-ins are managed, for instance, the *VDE* plug-in which is especially concerned in this section.

Area Selection. In general, images or single video frames (in fact also images) can be selected as areas to be annotated. As mentioned above, video frames are generated by means of the *Grab Frame* functionality provided by the *Video Player* component. In addition to that, internal areas of images may be marked, which can be performed manually or by applying an automatic feature. Manual selection is supported by the graphic drawing tools *rectangle*, *ellipse*, and *scribble shapes*. By default, an image or frame is automatically segmented into recognized regions when it is loaded. Users can then select single regions, but are also allowed to merge two or more areas by means of the *Magic Wand* tool. Finally, users need to select the ontology concepts which are assigned to media objects or areas and extracted descriptors as concept instance.

Information Attachment. Once the user has selected the required media area and semantic concept, the system conducts an automatic extraction of all previously indicated visual features. Visual features accord to the core MPEG-7 descriptors [Sik01] and are taken from a specific *Visual Descriptor Ontology* that is provided by the application. After extraction, the gathered visual descriptors are linked with the automatically created instances of the selected domain concepts. Hence, different types of information are generated: semantic information in form of concept instances, and descriptive metadata as visual features. Attributes and values assigned to concept instances can also be edited in the Ontology Browser.

Search and Exploration. With regard to search, no respective functionalities has been detected. The major part of navigation features are adopted in the *Ontology Browser*. Here, the structural information of a loaded ontology is presented in the *Class* panel. Concept instances are visualized in the *Individuals* panel, and their semantic relations are provided in the *Object Properties* area.

Import and Export. M-OntoMat-Annotizer is able to import and export ontologies as RDF(S) ontology files. In the case of export, merely domain instances and assigned visual descriptors are included, while the original ontology remains unmodified. Additionally, reports of the automatic extraction process can be generated, which are displayed in a separate window in textual format.

Process Assistance. Two essential automatic approaches are provided that release human effort at relevant parts of the annotation process: automatic segmentation of images or video frames into shape areas, and extraction of visual features according to core MPEG-7 descriptors.

3.2.9 Music Annotator

Music Annotator is a GPL (GNU General Public License) scientific tool for the analysis and annotation of musical data, developed at the Musical Technology Group at Universitat Pompeu Fabra of Barcelona, Spain [AAG06, Cla]. Here, the main objectives are to visualize, examine, verify, and modify information extracted from audio tracks. Among this information are low-level features, note segmentation, chords, or track structure. Furthermore, it permits manual annotation as training data or ground truth with regard to the applied extraction algorithms. Music Annotator is part of the so-called *CLAM framework* which enables a flexible system extension by integrating additional extraction algorithms. Interoperability is also fostered by storing and managing all data in XML format. Music Annotator accepts OGG, MP3, WAV, and AIFF audio file formats.

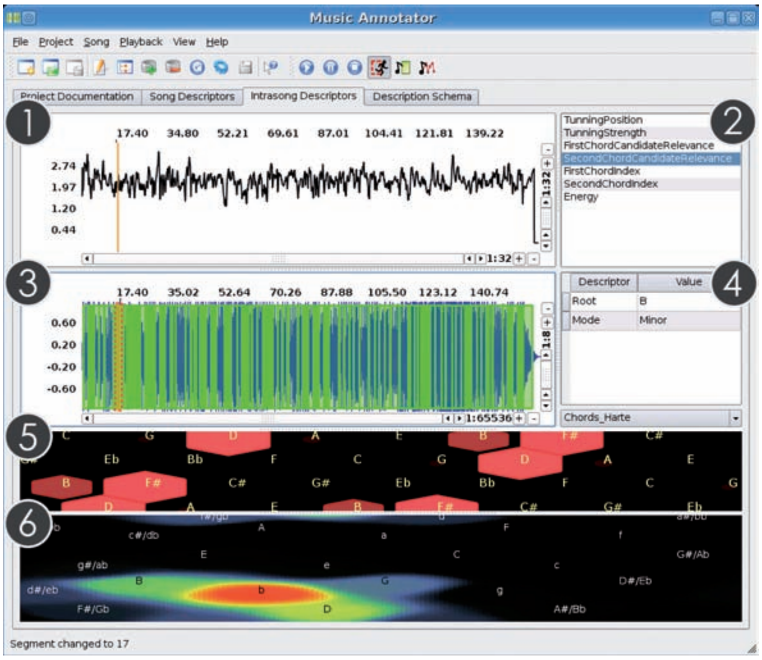


Figure 3.9: User Interface of CLAM Music Annotator. [Cla]

The graphical user interface of Music Annotator is subdivided as follows (see Figure 3.9). The top panel (1) is responsible for the visualization of numerical values that are extracted from the chords of a music track. The respective set of referred attributes is represented in the right-hand panel (2). Beneath, the waveform of the opened track is visualized (3). Upon the waveform, detected chord segments are highlighted. To the right, attributes and attached annotations referring to these chord segments are listed (4). The next panel named *Tonnetz View* (5) presents hexagons that represent sounding pitches. Here, hexagon size and color intensity indicate for the relevance of the related chord within the entire music track. On the bottom panel named *Keyspace View*, the instant tonality that matches with the major and minor chords is displayed.

System Configuration. By means of the *Project Properties* panel, basic project parameters and visual properties for the graphical user interface can be customized. Additionally, so-called *description schemes* can be specified and edited. A description scheme is a file that describes the working attributes of the project. It comprises all descriptors that are going to be loaded. Furthermore, a command can be inserted that is used to invoke a specific extraction tool.

Area Selection. After opening an audio file, chords along the waveform are defined as segments of the track. These are visualized in forms of color-marked wire frames. Chords refer to time segments (intervals) or time marks, i.e., they represent notes, break points, chord regions, or structural song parts. Since potential faults have to be expected regarding the automatic extraction, segments and marks may be subsequently edited. Here, users can move and resize the graphical representatives by means of Drag&Drop, and are also enabled to insert new elements by means of a specific keystroke-mouse interaction.

Information Attachment. First of all, the entire song track can be enriched with specific textual and numerical *Song Descriptors* such as *artist*, *title*, *genre*, *danceability*, *key*, *mode*, *dynamic complexity*, or *bpm* (beats per minute). After the automatic chord extraction, further descriptors are assigned to the resulting chord segments (which also can be modified later on). Annotated descriptors refer to the specified description scheme and can be classified as low-level and high-level descriptors. A description scheme contains two kinds of information: *attributes* and *scopes*. Attributes are value holders and obtain a specific validity that is limited to a given scope. For instance, the attribute "pitch" is valid within the scope "note". By means of the *Aggregation Extractor Tool*, identified descriptors of different extractors can be combined. In addition to that, it is possible to include multiple extractors such as a low-level spectral descriptor extractor, a ID3[6] descriptor extractor, or a chord extractor that extracts segments labeled with chords.

Search and Exploration. In general, navigation is enabled at multiple levels according to the different panels of the application. These levels can be divided into general song descriptors and song-subset descriptors. On the latter level, annotations (descriptors) can be explored for each selected visual element, e.g., for single chord segments of the waveform view. Regarding search queries on the given data set, no respective functionalities have been identified.

Import and Export. Music Annotator stores and manages any data in a XML-based format. In this way, interoperability and the orientation on established standards is guaranteed. Thus, the system generates interpretable files that can be connected to external applications or databases.

Process Assistance. The annotation process is assisted by the integration and potential combination of several automatic tools for track segmentation and descriptor extrac-

[6] ID3 is a specific metadata format for MP3 tracks. For more information see http://www.id3.org/.

tion, which replace human effort at specific points. Additionally, basic keystrokes are defined, e.g., for manual segment creation purposes.

Collaboration. A teamwork-based annotation of larger music track collections is enabled by the employment of a connected *BOCA* framework (Backbone of Collective Annotations). Additionally, a BOCA client is integrated. Within a BOCA server, a set of songs that are to be processed as well as tasks can be specified such as “insert bpm”, “insert ID3 descriptors”, etc. These specifications are presented in forms of a to-do list. By performing queries to web services, the included BOCA client reads that list out and automatically creates a Music Annotator project. All modifications conducted by the individual user are sent back to the framework, and the respective item of the to-do list is marked as “accomplished”.

3.2.10 Vannotea

Like *AnnoCryst* (see Section 3.2.3), a further reference implementation of *Annotea* [Annb] is *Vannotea*, developed at the Distributed Systems Technology Centre of the University of Queensland, Australia [UoQI]. The system allows collaborative groups to annotate, analyze and perform discourse around images, video, audio, and - in junction with *AnnoCryst* - 3D models [SHG$^+$06]. In particular, teams of trusted colleagues within research or academic environments are addressed. Collaboration is aimed to be supported in two ways. By means of one or more Annotea servers, annotations and contents that result from asynchronous teamwork can be shared and centrally stored. Second, synchronous communication, event-logging, and application sharing is enabled by an integrated Jabber-Client. Vannotea supports annotation of the following media and file formats: HTML, images (JPEG2000), a majority of video and audio formats depending on the running media platform, as well as 3D-Models (accompanied by *AnnoCryst*).

As pictured in Figure 3.10, the main components of the Vannotea workspace are an *Embedded Web Browser* (1), *Embedded Media Players* (2), an *Annotea Sidebar* (3), as well as a panel that offers different visualization and communication options (4). By means of the *Embedded Web Browser*, access is granted to existing audio-visual archives that can be explored through common search and navigation services. One or more *Embedded Media Players* display annotated media documents of different supported formats. The *Annotea Sidebar* is the central element referring to search and exploration of the set of generated annotations. The lower part of the user interface includes an *Annotea Timeline*, *Jabber Client*, *Jabber Chat*, and a *Record & Replay* tool.

System Configuration. Among others, relevant customization options are the selection of one or more content providers from which the integrated media documents are gathered. Further on, access control policies can be specified by users that are regarded as “first author” of a collaboratively annotated set of media.

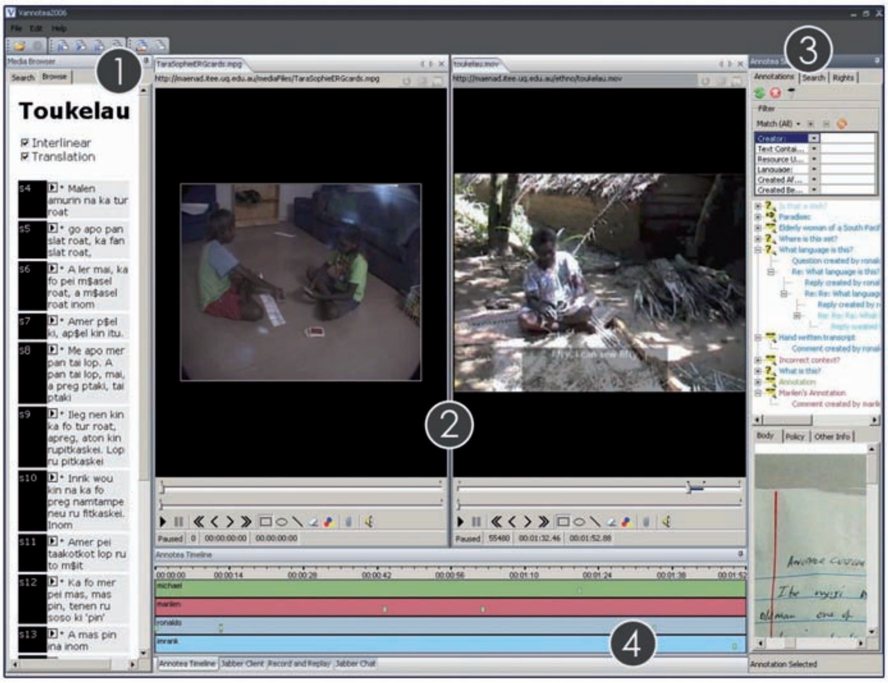


Figure 3.10: Multimedia Annotation with *Vannotea*. [UoQI]

Area Selection. For any kind of supported continuous media, time intervals can be defined by means of common mouse interactions. This can be done either on the respective media player or on the timeline view. In addition to temporal selection of areas of interest, Vannotea provides drawing tools by which graphical objects can be inserted on top of media. This option is used in the context of real-time collaboration and communication, at which co-annotators are allowed to point on or highlight parts of the media they are referring to. Drawings are stored as *SVG* file (Scalable Vector Graphic).

Information Attachment. Different kinds of additional information may be attached to selection areas. These include terms that are derived from (i) controlled vocabularies, (ii) free-text comments such as personal remarks, interpretations, or questions, as well as (iii) links and paths that point to external web resources or files in the local file system. Suchlike referenced files can be images, video or audio recordings, PDF files, etc. As it is the case with *AnnoCryst* (since both environments are based on the common assisting framework *Annotea*), each annotation may be additionally categorized by attaching metadata such as *author*, *date* of creation, which *resource* it annotates, *context* within the resource (for instance, as XPointer[7] to a HTML paragraph), or the annotation *type*. The latter may be specified as *Comment*, *SeeAlso*, *Question*, *Explanation*, *Advice*, etc.

[7]XPointer is a W3C standard query language that supports addressing of parts of an XML document. For more information see http://www.w3.org/TR/xptr-framework/

Search and Exploration. The integrated *Annotea Sidebar* serves as central unit in order to manage, explore, and conduct queries within the given data set. In forms of a tree view, own and other users' annotations and their content are visualized in hierarchical structure. From here, also new annotations can be created as well as removed and modified. The viewer also provides text-based search functionalities in order to find annotations, at which a search query is applied on users (names), media selections, annotation metadata and content, as well as *childOf* and *replyTo* relations. Search results are displayed in a specific list that presents segments (media selections) from different media objects which refer to the quested annotation. Another access point to the given information space is provided by the *Annotea Timeline*.

Collaboration. In general, collaboration is supported by means of a connected *Annotea* server that is employed in order to store and manage all data generated or modified in the course of annotation processes. Asynchronous communication is realized by means of the facilities of replying to existing annotations which are offered by the integrated *Annotea Sidebar*. On the other side, also synchronous collaboration and communication can be established. Here, other users are invited to an annotation session by a "first author", who is allowed to determine specific access policies that are assigned to the respective content. As already described, users are allowed to highlight parts of the content using graphical drawing tools, in order to point on the content she or he is recently referring to. Hence, browsing, discussion, and annotation can be conducted in real-time collaboration. Besides of the features provided by the *Jabber Chat*, synchronous communication is supported by video and audio conferencing facilities that are offered by the integrated *Jabber Client*. Moreover, the *Record & Replay* tool enables users to replay a collaborative session on the *Timeline* at a later time, at what activities of participating users are tracked and stored as event logs during a session.

3.3 Comparison and Subsumption of identified Features

In the previous sections annotation-related features of different digital media annotation systems were identified and assigned to inductively developed categories. Here, the categories *System Configuration*, *Area Selection*, *Information Attachment*, *Search and Exploration*, *Import and Export* relate to sub-processes of annotation, and *Process Support* and *Collaboration* consider functionalities provided in order to support annotation workflows and collaborative work. In the following, the exemplified applications will be compared by contrasting and subsuming detected features according to the introduced process-related categories.

3.3.1 System Configuration

With respect to application configuration, Table 3.1 reveals that two classes of administrative features have been identified: customization and management of users and data involved in the annotation process, as well as customization and storage of general settings.

Configuration											
		A.nnotate	ATLAS.ti	AnnoCryst	ELAN	EVA	EXMARaLDA	MADCOW	M-Onto.-Annot.	Music Annotator	Vannotea
1.	Customization and management of users and data	±	•	•	±	•	−	−	−	−	•
2.	Customization and storage of general settings	•	•	±	•	•	•	•	•	•	•

• Explicitly supported.
− Not or insufficiently supported.
± Implicitly supported or limited to single features.

Table 3.1: Configuration Features of Multimedia Annotation Systems.

(1.) The first item includes an account-based management of users that are stored in specific user databases, as it is provided by ATLAS.ti. Relations between users and the processed data are described by the definition of account-related access rights. Examples are comprehensive rights management (AnnoCryst, Vannotea) and assignment of contents to single users as so-called workload (EVA). Concerning resources of handled data, AnnoCryst and Vannotea allow the indication of data providers of which media sources are obtained. In the same context, ATLAS.ti realizes data source management. Furthermore, annotation schemes such as vocabularies, descriptions schemes, and categorization systems (ELAN, EVA, EXMARaLDA, Music Annotator), or more structured semantics and ontologies (AnnoCryst, M-OntoMat-Annotizer, Vannotea) can be selected. In some cases, these schemes also can be edited within the environment.

(2.) The second field refers to general application settings and may include system and document properties, as well as visual configuration of the graphical user interface. Among system and document properties are management of integrated tools, for instance, plug-in management and connection to servers (M-OntoMat-Annotizer), or selection of available media players (EXMARaLDA) or extraction tools (Music Annotator). A.nnotate and ATLAS.ti allow the specification of document properties, e.g., whether it is private or shared. Further system properties can be the annotation language or method (ELAN), or customizations for data export (EXMARaLDA). User interface configurations refer on the visual representation of data and interface elements. For example, the size or color scheme of annotations may be defined (A.nnotate, ATLAS.ti, ELAN), or, it can be specified whether annotations are shown or hidden (MADCOW). This point is also applicable to user interface components such as windows or panels, specifying their size, color, position (ELAN, Music Annotator), or the number of displayed elements of a specific type (EVA).

3.3.2 Area Selection

Several functionalities have been identified with respect to area selection, that is, parts of a document that are specified as regions that can be annotated. Different types of selection areas can be exposed. An overview is provided in Table 3.2.

Area Selection		A.nnotate	ATLAS.ti	AnnoCryst	ELAN	EVA	EXMARaLDA	MADCOW	M-Onto.-Annot.	Music Annotator	Vannotea
1.	Entire document as content container	•	•	–	±	–	•	–	•	?	?
2.	Separated or embedded media object	•	•	–	•	•	•	•	•	?	?
3.	Multiple media objects	–	±	–	–	•	–	–	–	–	–
4.	Spatial region within media object	•	•	–	–	–	•	•	–	•	•
5.	Temporal region within media object	–	•	•	±	•	–	±	•	–	•
6.	Spatio-temporal region within media object	–	–	–	–	–	–	±	–	–	±
7.	Multiple types of areas-to-annotate (segments)	•	•	–	–	–	–	•	•	±	•
8.	Multiple ways of selecting areas-to-annotate	•	•	–	•	±	•	•	•	±	•

• Explicitly supported.
– Not or insufficiently supported.
± Implicitly supported or limited to single features.
?: Information not available.

Table 3.2: Area Selection Features of Multimedia Annotation Systems.

(1.) First, annotations may be attached on the entire document level. This is featured by the majority of the systems, for instance, A.nnotate supports the selection of shared

DOC or PDF files. By means of screenshot or snapshot features, images of web pages (A.nnotate) or GoogleEarth excerpts (ATLAS.ti) can be generated and defined as areas of interest.

(2.) Media objects, either separately or embedded within a document, can also be selected, mostly through mouse click (A.nnotate, EVA, M-OntoMat-Annotizer).

(3.) Also multiple media objects, e.g., contained in one single media document, can be selected and annotated in simultaneous manner. For instance, EVA allows selecting of thumbnails for equal annotation.

(4.) Going further, also parts within a media object such as spatial regions may be marked. Drawing tools can be employed in order define graphical forms, e.g., to highlight regions within images or single video frames (MADCOW, M-OntoMat-Annotizer) that can be stored in a graphics format such as SVG (Vannotea). Both A.nnotate and ATLAS.ti enable users to mark text passages as sequence of characters, or to draw bounding boxes upon a document. By means of XPointers, paragraphs can be addressed within an XML document or an HTML file (Vannotea). AnnoCryst supports the selection of objects that are part of a 3D-Model.

(5.) In the case of continuous media such as audio or video, temporal areas may be specified. Here, time intervals or points are supported by ATLAS.ti, Music Annotator, ELAN, EXMARaLDA, and Vannotea. Additionally, spatial information can be combined with temporal selections forming spatio-temporal areas.

(6.) Implicitly, definition of spatial regions on single video frames, combined with a point in time (M-OntoMat-Annotizer) or a time interval (Vannotea) is supported. That is, so-called spatio-temporal areas can be specified.

(7.) Concluding, Table 3.2 shows that five of the examined systems explicitly support multiple types of selection areas (see 1. to 6.). One application implicitly does on a rudimentary level (Music Annotator).

(8.) There are also different ways of selecting annotation parts. The most frequent interactions are Drag&Drop, which are featured by all systems. Here, activities are fostered such as text marking, drawing boxes and graphical objects, define time intervals, or select more than one media object. Another form, single mouse clicking, can be conducted either upon the referred media object (AnnoCryst, EVA) or on a button, e.g., in order to define the boundaries of a time interval (MADCOW). Interval boundaries may also be defined by entering values in a respective input field, as it is realized by ATLAS.ti and EXMARaLDA. Furthermore, specific keyboard operations are means of inserting time-based segments (ELAN, Music Annotator), or to define start and endpoints of a time interval (EXMARaLDA). As the only system, EXMARaLDA allows real-time selection, i.e., the specification of time interval boundaries during an audio or video track is playing. Besides of these forms of manual area selection, suchlike activities are assisted by the application of automatic approaches. Music Annotator provides tools that automatically segment an audio waveform. Also

EXMARaLDA supports audio segmentation, whereas this data is only exported and not used during the annotation process. M-OntoMat-Annotizer conducts an automatic region segmentation on images and video frames. As illustrated in Table 3.2, nine of the ten annotation systems offer more than one feature in order to perform selection.

3.3.3 Information Attachment

Information Attachment refers to features that are provided in order to insert or connect additional information to the original media content. The essential aspects are illustrated in Table 3.3.

Information Attachment											
		A.nnotate	ATLAS.ti	AnnoCryst	ELAN	EVA	EXMARaLDA	MADCOW	M-Onto.-Annot.	Music Annotator	Vannotea
1.	Specification of low/ high level metadata	•	•	•	•	•	•	•	•	•	•
2.	Entering free-text	•	•	•	•	–	•	•	–	•	•
3.	Assignment of (multimedia) content	•	±	•	–	–	•	•	–	–	•
4.	Creation of hyperlinks, relations, or networks	–	•	±	•	–	•	±	–	–	±
5.	Multiple types of attached data	•	•	•	•	±	•	•	•	•	•
6.	Multiple ways of data assignment	–	•	±	•	–	•	–	–	•	±

• Explicitly supported.
– Not or insufficiently supported.
± Implicitly supported or limited to single features.

Table 3.3: Information-attachment Features of Multimedia Annotation Systems.

(1.) It has been detected that all analyzed tools provide the annotation of metadata of low or high level. Low-level metadata includes textual descriptive data such as color (M-OntoMat-Annotizer), or numerical descriptions like beats per minute (Music Annotator). Low-level metadata can also be tags (freely defined categorical terms), provided by A.nnotate and AnnoCryst. In this context, terms of descriptive and categorical metadata can be obtained from predefined schemes such as (controlled) vocabularies or category systems (ATALS.ti, ELAN, EVA, Music Annotator). Classification on a higher level is enabled by semantic annotation, that is, terms or concepts that derive from a semantic structure or ontology (AnnoCryst, M-Ontomat-Annotizer). Another form of metadata is information that is attached to an annotation, giving additional description as *metametadata*. These may include type, author, title, creation or modification date, location, or context, and are supported by AnnoCryst, EVA, ELAN, EXMARaLDA, MADCOW, and Vannotea.

(2.) Eight of ten investigated applications allow free-entering of text that can be exploited in order to phrase comments, questions, interpretations, or opinions. EXMARaLDA also supports the addition of special characters that arise from the International Phonetic Alphabet. Furthermore, when annotation on annotations are enabled, textual free-text is to be regarded as fundamental prerequisite of collaborative discourse (A.nnotate, AnnoCryst, ELAN, MADCOW, Vannotea).

(3.) Not only textual or numerical metadata, but also multimedia content of several types can be annotated, mostly by means of inserting a link to a web resource or a path within the local file system (A.nnotate, AnnoCryst, EXMARaLDA, MADCOW, Vannotea). Here, an unlimited number of different media file formats are conceivable.

(4.) Linking is also means of creating relations between elements of an annotation project such as media files, selection areas, or annotations themselves. For instance, related features are provided by ATLAS.ti, ELAN, and EXMARaLDA.

(5.) Table 3.3 demonstrates that all annotation systems support multiple types of annotated data, either as low-level or high-level metadata, textual or multimedia content, or as relations.

(6.) Six out of ten systems provide more than one method for data assignment. Consequently, not only the provision of different *types* of annotation can be adhered, but also different *ways* of generating and attaching additional information.

3.3.4 Search and Exploration

Search and Exploration functionalities are applied in order to support users during information seeking, navigation and browsing within the entire data set which includes media files, annotations, other users, and external resources. The different asprects related to Search and Exploration activities are summarized in Table 3.4.

(1.) As illustrated in Table 3.4, one of fundamental activities of exploration is searching. With respect to search queries in order to find elements of an annotation project, the most basic feature is keyword or text-based search, at which entire documents or sets of annotated data can be examined (A.nnotate, ATLAS.ti, AnnoCryst, ELAN, EXMARaLDA, MADCOW, Vannotea).

(2.) Advanced search options are provided. These include pattern matching, category search, or combinations-based search of annotations (ATLAS.ti), semantic SPARQL queries (AnnoCryst), specification of areas that searches apply to (EXMARaLDA), or the supply of search histories, query storage, or export of search results (ELAN).

Search & Exploration											
		A.nnotate	ATLAS.ti	AnnoCryst	ELAN	EVA	EXMARaLDA	MADCOW	M-Onto.-Annot.	Music Annotator	Vannotea
1.	Search functionalities (e.g., keyword-search)	●	●	●	●	–	●	●	–	?	●
2.	Advanced search options	–	●	●	●	–	●	–	–	?	●
3.	Search and retrieval from different resources	–	–	●	●	–	±	●	–	±	●
4.	Multiple forms of wayfinding	●	●	●	●	●	●	–	–	●	●
5.	Creation of groups and hierarchies	●	●	●	●	●	●	–	±	–	●
6.	Bookmarking, storage of outstanding content	±	–	–	–	–	●	–	–	–	–

● Explicitly supported.
– Not or insufficiently supported.
± Implicitly supported or limited to single features.
?: Information not available.

Table 3.4: Search and Exploration Features of Multimedia Annotation Systems.

(3.) Not only the data contained in one project, but also external resources may be involved in queries. Thus, AnnoCryst allows seeking for 3D-Models in scientific databases, or semantic SPARQL queries on external resources. Vannotea grants access to multimedia web resources. In a less global manner, ELAN supports search queries on multiple project files.

(4.) Indeed, all systems provide possibilities to explore the data set by navigation. Here, eight applications offer more than one method for wayfinding. For example, ATLAS.ti, ELAN, Music Annotator, and Vannotea include different access points from various windows or panels. Among these representations are simple lists, tree-visualizations, and timelines. Upon documents, MADCOW displays icons that represent annotated content. Step-by-step navigation through media, selections, pages, etc., is provided by ELAN, EVA, and EXMARaLDA. Further forms of structured data representation are tag clouds and graph-based visualizations (AnnoCryst). Another level of exploration is constituted by manipulation, filtering, and sorting, which can be applied to media files and annotations as well. Media files such videos or audio tracks can be navigated through various control elements (ATLAS.ti, EVA, EXMARaLDA, MADCOW, Music Annotator, M-OntoMat-Annotizer, Vannotea) or audio waveforms through pan and zoom (EXMARaLDA), and also 3D-Models may be explored by the application of rotation, pan, and zoom facilities (AnnoCryst). AnnoCryst and ELAN allow observation and exploration of annotated media objects from different views by realizing multiple displays. For representation manipulation purposes, ALTLAS.ti provides filtering (also definition of own expressions) and element shifting through Drag&Drop. A.nnotate allows user to choose between different ways of displaying documents and annotated content.

(5.) Manipulation of the data representation is also means of creating groups and hierarchies. A majority of the analyzed systems provide structural representation and organization of information in order to restructure information according to specific relations. ATLAS.ti, AnnoCryst, and Vannotea display annotated information in hierarchies. Grouping and Clustering is supported by ATLAS.ti (*tiers* and *families*) ELAN (*layers*), and EXMARaLDA (layers).

(6.) As the only system, EXMARaLDA offers bookmarking of parts within a media object (here, audio track). Thus, contents that previously have been defined to be of special interest can easily be accessed and navigated afterwards.

3.3.5 Import and Export

Data import and export is an instrument to ensure cross-platform interoperability and uniformity of processed data. For this purpose, several formats have been identified that can be transfered to or from external applications or services (see Table 3.5). The most significant features have been identified in the field of audio processing and transcription, which is assisted by ATLAS.ti, ELAN, and EXMARaLDA. Here, file formats of annotated data from familiar applications can be imported and exported as well. Among these applications are, to name but a few, *CHAT*, *Praat*, *Shoebox*, or *TASX* (cf. appendix A).

(1.) In order to work with classification schemes, different levels of descriptive metadata can be imported as templates, e.g., controlled vocabularies (ELAN) or RDF ontologies (AnnoCryst, M-OntoMat-Annotizer, Vannotea). Also results from certain parts of previously conducted annotation processes may be integrated. EVA imports pre-segmented video (detected shots) that are coded as MPEG-7 files.

(2.) Diverse export formats and results have been localized. These include PDF files (A.nnotate), HTML and XML (ATLAS.ti, ELAN, EXMARaLDA, Music Annotator), SMIL (ELAN) , or RDF(S) (M-OntoMat-Annotizer). Also text-based formats are exported, for instance, CSV Comma Separated Values (A.nnotate), Tab-delimited text (ELAN), or RTF rich text (EXMARaLDA).

(3.) Another area of data export is the generation of summaries or reports. In this context, ELAN supports extraction of specific video parts, and printing of annotations within one single document. The latter is also featured by EXMARaLDA, which also exports HTML with an embedded Flash Player in order to present annotations and the

Import & Export		A.nnotate	ATLAS.ti	AnnoCryst	ELAN	EVA	EXMARaLDA	MADCOW	M-Onto.-Annot.	Music Annotator	Vannotea
1.	Import of externally annotated data	–	•	•	•	•	•	–	•	–	–
2.	Export to standardized file formats	•	•	–	•	?	•	–	•	•	–
3.	Generation of reports and summaries	±	±	–	•	–	±	–	±	–	–

• Explicitly supported.
– Not or insufficiently supported.
± Implicitly supported or limited to single features.
?: Information not available.

Table 3.5: Import and Export Features of Multimedia Annotation Systems.

related media in one common file. In textual format, M-OntoMat-Annotizer generates reports about the data resulting from the automatic extraction of visual descriptors.

3.3.6 Process Support

The least represented field of system features is process support. However, a consideration of this category helps to detect how the process has been supported by multimedia annotation systems up to now. As illustrated in Table 3.6, early stage functionalities have been localized.

(1.) Only offered by EVA, monitoring of the process state is a means of, for example, checking if all tasks or all files previously defined to be processed are finished. Here, a so-called workload can previously be defined.

(2.) The most frequent feature concerns the use of keyboard strokes or shortcuts in order to accelerate selection and information attachment processes as well. This is explicitly supported by ELAN, EVA, and EXMARaLDA.

(3.) Another functionality is sequencing of successive steps, that is, automatically starting an activity or displaying a tool after a previous activity has been accomplished. Within the set of analyzed systems, suchlike features are supported only on a rudimentary level, for instance, an annotation dialog is displayed after selecting an area.

(4.) Finally, the annotation process can be fostered by allowing real-time annotation, i.e., performing activities during media-playback. EXMARaLDA enables users to define start and endpoints of time intervals while a video or audio file is playing.

Process Support		A.nnotate	ATLAS.ti	AnnoCryst	ELAN	EVA	EXMARaLDA	MADCOW	M-Onto.-Annot.	Music Annotator	Vannotea
1.	Monitoring of process state	–	–	–	–	●	–	–	–	–	–
2.	Use of keyboard strokes or shortcuts	–	±	?	●	●	●	–	–	±	?
3.	Sequencing of successive steps	±	–	–	±	±	–	–	±	–	–
4.	Real-time annotation during media display	–	–	–	–	–	±	–	–	–	–
5.	Simultaneous processing of multiple media documents	–	●	–	●	●	–	±	–	–	–
6.	Integration of automatic approaches	–	±	–	±	±	±	–	●	●	–

● Explicitly supported.
– Not or insufficiently supported.
± Implicitly supported or limited to single features.
?: Information not available.

Table 3.6: Process-support Features of Multimedia Annotation Systems.

(5.) In addition to that, the possibility of processing multiple documents at the same time seems to be useful, since completeness can be assisted if larger sets of media objects need to be annotated (ATLAS.ti, ELAN, EVA).

(6.) Time-consuming manual annotation can be countered by the application of automatic processes. In this context, Music Annotator supports automatic audio waveform segmentation and extraction of low-level features such as ID3 descriptors. M-OntoMat-Annotizer provides tools that automatically segment image regions, and extract previously selected MPEG-7 visual descriptors.

3.3.7 Collaboration

As illustrated in Table 3.7, the general collaboration-related functionalities refer to facilities of task distribution, communication, content sharing on different levels, and awareness of teamwork.

(1.) Two forms of task distribution have been localized. EVA allows the assignment of media files to individual users and workload shifting during runtime. Server-based, Music Annotator puts to-do lists at users' disposal, at which individual users may download and undertake tasks and send modifications back to the server application.

Collaboration											
		A.nnotate	ATLAS.ti	AnnoCryst	ELAN	EVA	EXMARaLDA	MADCOW	M-Onto.-Annot.	Music Annotator	Vannotea
1.	Account-based task and range assignment	–	±	±	±	•	–	–	–	•	±
2.	Communication and Discussion	•	–	•	•	–	–	•	–	–	•
3.	Content Sharing	•	±	•	•	•	–	•	–	–	•
4.	Centralized data storage and access	•	±	•	–	•	–	•	–	•	•
5.	Modifications on the same data	±	–	•	–	•	–	•	–	•	•
6.	Collaborative Awareness	–	±	•	•	±	–	–	–	±	•

• Explicitly supported.
– Not or insufficiently supported.
± Implicitly supported or limited to single features.

Table 3.7: Collaboration Features of Multimedia Annotation Systems.

(2.) Communication is supported in two different ways. Asynchronous communication is realized by allowing users to reply on existing annotations, mostly in textual form (A.nnotate, AnnoCryst, MADCOW, Vannotea). Not only text, but also further media formats can be used as replying annotations, for instance, comments recorded on video (MADCOW). In the scope of synchronous communication, chat tools are frequently applied (AnnoCryst, ELAN, Vannotea). Additionally, Vannotea also provides video and audio conferencing components.

(3.) Content sharing regards the general access to public content, which includes media documents and annotations as well. Sharing may only be realized by working on copies and sharing through content forwarding via email. Accordingly, content sharing in general is supported by A.nnotate, AnnoCryst, ELAN, EVA, MADCOW, and Vannotea.

(4.) Including sharing in general, content may also be stored on a central unit, at which a physical distribution of large collections of content is avoided. For this purpose, client-server-architectures are implemented (A.nnotate, AnnoCryst, EVA, MADCOW, Music Annotator, Vannotea).

(5.) Collaboration is often regarded as working on the same (physical) content. Here, working on the same content is not a essential requirement for the previous points *sharing* and *centralized storage*. Modifications to the same data is allowed to (authorized) users by AnnoCryst, EVA, MADCOW, Music Annotator, and Vannotea.

(6.) Collaborative awareness means that individual users working within a group obtain information about the presence of co-working users. According to that, AnnoCryst notifies user through RSS Feeds when other group members have made changes on a related data set. In the scope of synchronous collaboration, both ELAN and Vannotea allow co-annotators to point on content they want to discuss on or modify by means of highlighting or drawing. In addition to that, Vannotea also enables recording of a synchronous session, as well as a "replay" of recorded collaboration at a later time.

3.4 Summary and Conclusions

In this chapter, a collection of different features provided by (multi-)media annotation systems have been localized and specified, at which the term features includes the functionalities, tools and approaches provided by annotation systems in order to perform certain annotation tasks, as well as general forms of data which are generated and treated by these functionalities. Here, two general aims have been pursued. First, different forms of annotation and respective systems had to be considered, ensuring that created results include diverse aspects of multimedia annotation. Thus, a classification of annotation systems was conducted which based on a first inspection of more than eighty annotation systems. Thereafter, ten representative applications were chosen for detailed examination. Second, identified features had to be assigned to different activities or tasks which are contained in annotation workflows on the one hand, and to the general topics of this thesis on the other hand. For this purpose, a task-related category system has been exploited which was inductively developed in the course of the examination of single application. Thus, the function and purpose of features in the context of the entire system has been defined referring to specific work items within the process. According the defined classes of annotation tasks, the following results can be summarized:

- Features have been identified which can be assigned to administrative processes of *System Configuration*. Basically, such tools and functionalities deal with the management of user data and related information, as well as the customization of general project settings and the individual configuration of the user interface.
- The next class of features relates to the *Selection of Areas* which are connected with additional information, determining the context of annotations within the edited media documents. Among these areas might be the entire document, one or more single media objects, or specific spatial and/or temporal boundaries within a media file. For the latter case, structural metadata is mostly generated.
- Furthermore, features have been extracted with respect to the *Attachment of Information* to previously selected areas. Here, generated types of data can be descriptive or categorical metadata on a low level, higher-level descriptive or

categorical semantic information, different forms of media content, or links and relations.

- A further task-related group of features has been assigned to acts of *Search and Exploration*. These include basic or advanced search functionalities potentially from different resources, different forms of navigation and browsing, and facilities for manually structuring of given information, for instance, by defining groups and hierarchies. In addition to that, bookmarking has been detected as a useful feature in order to support long-term explorative processes.

- *Import and Export* tools have been localized which are to be regarded as means of cross-system interoperability, so that results can be delivered and reused by other applications which may provide additional functionalities. Particularly, it has been showed that standardized formats such as XML-based data structures are treated. Additionally, export tools are exploited which enable the generation of reports and summaries.

- According to the general research issues of this thesis, features have been examined with respect to *Process Support*. In general, these include process state monitoring, use of keyboard, sequencing of successive steps, processing of multiple files at the same time, and the provision of automatic approaches.

- In this work, also scenarios of *Collaboration* are taken into account. With regard to collaborative annotation workflows, fundamental features enable a distribution of the annotation workload among participators, which means both the single annotation tasks as well as the set of media documents to annotate. More general, collaboration support is provided by tools and functionalities for communication, data sharing and common access, and modification of the same physical data. Finally, some of the examined tools provide collaborative awareness, that is, individual users obtain information about other persons working in the same environment and project.

To sum up, essential knowledge has been created referring to different areas of annotation tasks and respectively applied tools, functionalities, or approaches, as well as assigned forms of data. These aspects constitute a basis for the definition of a *Generic Process Model of Multimedia Annotation* which will be presented in Chapter 5.

In addition to these central results, further conclusions can be drawn the regard to the area of problems and challenges emphasized in this thesis. First, this chapter revealed that in fact a *wide variety* is to be associated with the field of multimedia annotation systems. Accordingly, the following findings can be subsumed: More than half of the examined systems supports multiple types of selection areas. According to that, nine of the ten examined annotation systems offer more than one feature to perform selection. Moreover, all systems support multiple types of annotated data. Here, all of them allow annotation of low-level or high-level metadata. All systems that feature multimedia content attachment allow multiple media formats. In this context, six of the ten examined systems provide multiple ways of annotating data. Further on, several systems are able to import and export different types of data. Additionally, multiple forms of synchronous and asynchronous collaboration is supported in different ways. Second, a relevant finding is that little or no consideration is given to the *annotation process*, regarded as operational unit in which different tasks and activities are performed. Rudimentary process support, if any, is realized through features such as opening of a new window after a certain activity has been performed, use of keyboard in order to accelerate certain steps, or simultaneous handling of multiple media objects. As a conclusion, the relevance of the problem domain treated in this thesis can be reaffirmed.

While the examinations presented in this chapter especially focused on functionalities and forms of data, the next step is to regard the operational sequences which define the transitions between tasks and activities within the annotation process. For this purpose, two *Studies on Annotation Practices* were conducted in the scope of this work, whose results will be illustrated in the following chapter.

Chapter 4

Studies on Annotation Practices

As the main goal of this work is to foster processes of digital multimedia annotation, it is indispensable initial groundwork to obtain detailed knowledge about these processes in general. According to this, practices of annotation have to be investigated and specified, that is, it is to be clarified how people effectively work with annotation environments. While the analysis of multimedia annotation systems condcuted in Chapter 3 revealed application functionalities as well as resulting forms of data, this chapter focuses on the specific procedures and sequences performed by humans and machines during the annotation process. For this purpose, an empirical study on annotation practises was conducted (Section 4.1), at which experts from different research and educational institutes were interviewed and observed while operating the respectively applied annotation software. In Section 4.2, a survey of related literature is given which considers existing models on annotation workflows, field reports, as well as proposals for the realization of annotation projects. As a result, concrete workflows of annotation are derived by describing the comprised sub-processes and more granulated tasks, as well as their sequential interrelations. In Section 4.3, these use case specific workflows are compared and assembled, at which a futher basis for the definition of a *Generic Process Model of Multimedia Annotation* presented in Chapter 5 is constituted.

4.1 Empirical Study on Annotation Practices

Empiricism comprises the collection of information based on observations made in laboratory or field studies. Thus, knowledge is acquired by gathering information which relies on existing experience with regards to real objects and facts [TW]. Accordingly, the objective of the described study was to obtain knowledge about different annotation workflows, by investigating different examples of annotation which are (or have been) effectively applied in real application scenarios. To this end, experts at five research institutes were consulted. Here, the consulted persons where either teaching staff that instruct annotation, or professionals that annotate media content in everyday life. The institutes include the Leibniz Institute for Science Education at the University of Kiel, Germany (IPN), the Institute of General Education Science at the Technische Universität Darmstadt, Germany, the Knowledge Media Research Center in Tübingen, Germany (KMRC), the Telecooperation Group at the Technische Universität Darmstadt, Germany (TK), and the Institute for Sports Science at the Technische Universität Darmstadt, Germany.

4.1.1 Study Design

As stated above, the main goal was to detect annotation practices that are applied in the praxis. For that purpose, the meetings were structured into two parts. The first part included semi-structured interviews. A respective questionnaire basically referred to the individual use cases, annotation practices, and workflows including subprocesses and tasks, the tools and functionalities featured by the constituted software, and the users and roles involved in the annotation process. In this phase, the number of interviewed people differed from group sessions of four to six people, to individual talks with one expert. The applied questionnaire is illustrated in Appendix B.

In the second part of a meeting, one or two experts were observed while operating the annotation system employed in their usual praxis. The observation of software operation enabled the investigation of user activities and human-machine-interaction at certain phases or tasks of the annotation process. Here, a common annotation procedure was simulated (with realistic data), such as it is or has been performed by students or the experts themselves in real situations.

The conducted expert interviews and observations of system operation were lead by two persons. The first person was responsible for questioning and conversation, while the second person recorded the gathered information.

4.1.2 Practices of Annotation

Here, the results of the conducted interviews and observations will be presented for each consulted institute. First, general information about the respective institute is given. Afterwards, information is illustrated about (i) the setting in which annotation takes place, (ii) the goals and purposes of annotation, (iii) the types of resulting digital documents, and (iv) the annotation procedure. Essentially, different concrete workflows of media annotation are derived and presented.

Analysis of Lesson Recordings

The Leibniz Institute for Science Education (IPN) at University of Kiel, Germany conducts basic research with respect to learning issues, fostering and enhancing knowledge in the pedagogics of natural sciences and mathematics. In this context, the IPN conducted analyses of lesson videos recorded in the scope of the international *PISA Studies*, using the video annotation and coding software *Videograph* [Rim05, SPK05]. The general aim was to detect inadequacies, but also good examples of teaching, by analyzing the course of action and behavior of teachers during physics education lessons.

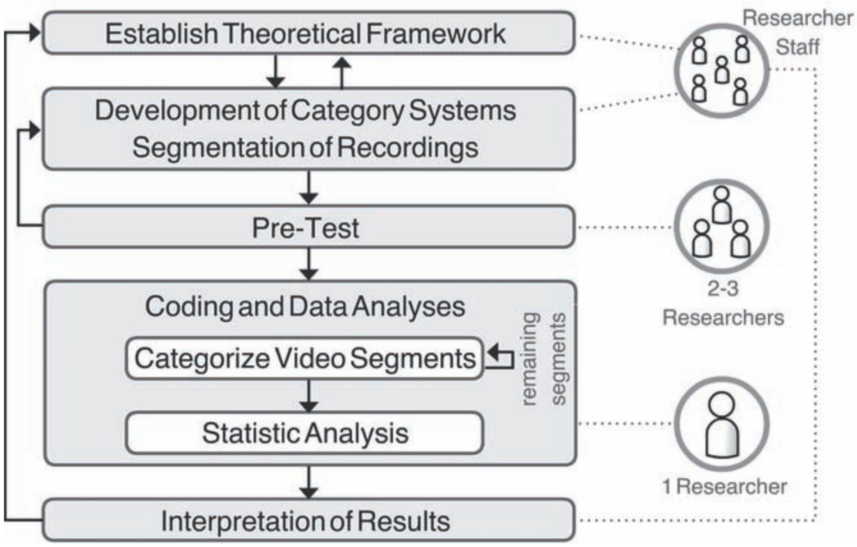


Figure 4.1: Phases of a video study conducted at the IPN Kiel (cf. [SPK05]).

As shown in Figure 4.1, the first part of the video study referred to the constitution of a theoretical framework. This included an identification of involved research areas and relevant research questions, as well as the formulation of a general hypothesis. Furthermore, the incorporated research data was determined, for example, the analyzed lesson recordings.

Among others, the analysis based on the categorization of video segments, at which the incorporated categories were obtained from predefined vocabularies (category systems). In the second phase, category systems were developed by operationalizing categories and formulating so-called *coding instructions*. Furthermore, this phase included the *segmentation* of involved video files, at which time intervals were specified by means of the *Videograph* software. Here, either fix time intervals or events (free intervals) were specified.

In the following step, a pre-test was performed. In this context, two or three researchers performed a regular categorization procedure independently from each other. The first goal of this phase was to validate the determined categories in order to modify them if necessary. A second objective was training of the participating researchers. Here, so-called reliability checks were conducted after the annotation phase. That is, it was investigated if the different researchers achieved an appropriate level of agreement with respect to their results in comparison. Mismatching results were afterwards discussed in group meetings in order to constitute accuracy.

After the training of annotators and validation of category systems, each annotator categorized the predefined video segments independently. Here, researchers assigned one or more categories to video segments, e.g., by means of predefined keyboard shortcuts. Once all time segments were classified, the resulting data was statistically analyzed.

The last phase of the video study comprised the interpretation of results. Here, researchers could interpret, reflect, or rate the classified observations by means of composing free text or using predefined rating scales. Finally, the results were discussed by the entire group on the basis of defined research questions.

Collaborative Text Authoring and Annotation

The Institute of General Education Science at Technische Universität Darmstadt, Germany focuses on education and training in connection with General Education and Vocational Education. In the scope of the lecture "pedagogy of new media", two use case scenarios are realized: collaborative questioning and commentary of lectures scripts, named *Interactive Lecture Note*, and collaborative writing of text documents based on specific tasks. In both scenarios, students are supported by the collaborative text annotation system *eMargo* [GGR+05]. In the following, only the second use case, namely *Group-eMargo*, is described.

The basic idea of the Group-eMargo online seminar is to assign certain tasks to students which have to accomplish these tasks in team work by means of the *eMargo* software. The time frame exceeds one semester and, with regards to the treated contents, tasks relate to the topics of the attended face-to-face lecture. The main learning target is to acquire new knowledge by dealing with determined subjects and debate the involved issues. The results of an online seminar are several text documents generated in collaborative authoring work. Management of tasks and user accounts is realized by a *Moodle* task module [Moo]. *EMargo* is used for creation, editing, and annotation of text documents.

At the beginning of a seminar, groups of five students are formed, and user accounts with respective group assignments are created (see Figure 4.2). Next, each group receives six tasks, at which tasks are connected to a deadline, managed by the *Moodle* module. The first task is organizational. Students groups have to schedule the seminar procedure for the entire semester. Furthermore, each student is declared "first author" for one of the remaining five tasks. In doing so, tasks are assigned to one person that is responsible for quality of results and deadline compliance. In oder to accomplish the first task, students use a specific discussion tool provided by *eMargo*.

In the following steps, the tasks as regards content are edited. Here, one text document

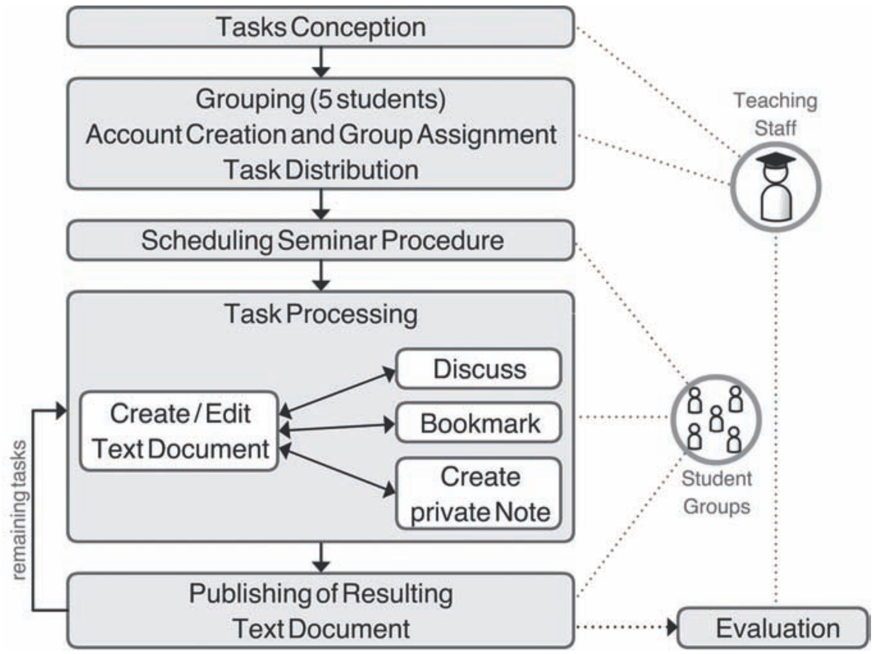


Figure 4.2: Process of the Group-eMargo Online Seminar.

is created and edited for each task, at which all students have to write a certain part of the document. For arrangement and agreement purposes, the integrated discussion tool can be used. Such contributions are to be regarded as annotations that can be replied on, and may be connected not only to the entire text, but also to paragraphs within the document. In addition to that, parts of the text can be bookmarked by means of a *Flag* tool. Students are also enabled to annotate private notes that are not visible to fellow students.

Once a task is accomplished, the first author generates a DOC file and uploads it to the *Moodle* task module, so that all published documents can be evaluated by the teaching staff at a later time.

Collaborative Hypermedia Design

Research at the Knowledge Media Research Center (KMRC) in Tübingen, Germany focuses on knowledge acquisition and knowledge communication by means of digital technologies. Traditional forms of teaching are investigated which refer to school and university education, as well as informal learning settings on the Web or workplace. In the scope of this research work, also university courses on collaborative design of hypermedia documents are performed. Among others, the video annotation and analysis system *WebDIVER* is employed [PMR$^+$04]. The workflow described in this section refers to regular university courses conducted at the University of Muenster, Germany, in cooperation with the Computer Graphics Center in Darmstadt, Germany

[SFZ06]. The main goal was to instruct students to collaboratively schedule and create hyperstructures of linked (self-shot) videos, at which videos also had to be attached with additional textual information. In this scope, principles of learning-by-design and project-based learning were realized (cf. Section 2.1.6). The courses were part of the psychology masters program. Accordingly, topics of the created hypermedia documents were “techniques of presentation and moderation”, “information system about study of psychology”, and “conflict management”.

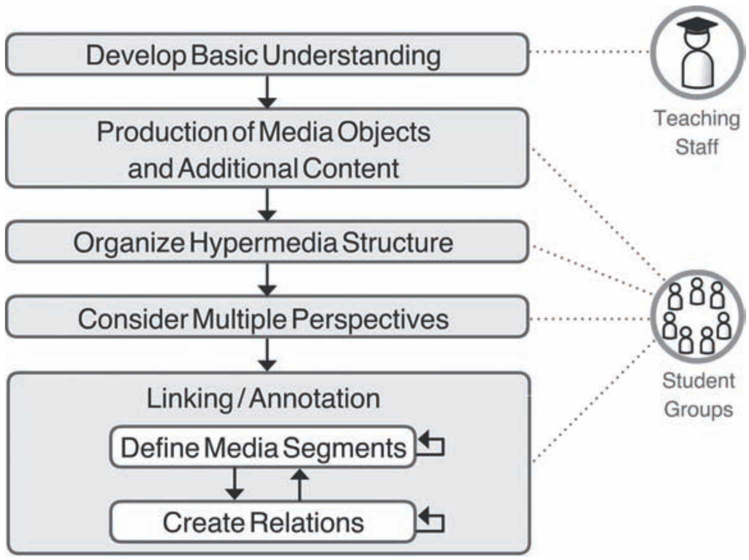


Figure 4.3: Collaborative Annotation at University Courses on Hypermedia Design. [SFZ06]

As illustrated in Figure 4.3, the annotation process was divided into five different instructional units, each covering a different aspect related to the hypermedia design process.

In the first phase, students should obtain a basic understanding of the characteristics of hypermedia design. In particular, this included mediation of the concepts of hypermedia. Furthermore, based on so-called *concept map videos* (a visualization of the planning phases) students were guided to plan and revise the materials, create storyboards, communicate organizational issues by means of a chat tool, and distribute tasks and responsibilities.

Next, students had to determine the contents and produce the incorporated videos and additional texts. Here, decisions were required about the media format in which information should be coded, and the amount of information that should be pooled in one information unit (e.g., textual information should only refer to a specific part of the video content).

In the following phase, the overall structure of the hypermedia documents was scheduled. Here, the macrostructure of the documents had to be determined. This included the structure within single videos concerning its attached additional information on the one hand, and the overall structure of the hypermedia documents referring to the relations between different video files on the other hand.

Moreover, students were instructed to define further relations, considering multiple perspectives of the presented content. For that purpose, three different alternatives were provided: First, different perspectives could be presented through the videos, e.g., by including several camera perspectives. Second, different additional text documents could be annotated to the same part of a video. Additionally, on the overall structure level, students were able to design different paths, for instance, different guided tours or structural overviews.

The last unit of the annotation process comprised planning and creation of video segments and relations. At video segmentation, so-called *sensitive regions* were defined. Sensitive regions can be regarded as spatio-temporal selection areas, which are wireframes that "follow" video objects with respect to their position and size in the video display area during playback time. In addition to that, relations were specified as hyperlinks between video segments on the one side and either other videos, video segments, or additional text nodes on the other side.

Annotation of Paper-based and Digital Documents

The main research area of the Telecooperation Lab at Technische Universität Darmstadt, Germany is ubiquitous computing or ambient intelligence. Among others, the focus is on networks and distributed systems as well as multimodal interaction techniques. Here, *CoScribe* was developed, a prototype application for paper-based and digital document annotation and sharing within small communities [SBM08]. By means of a digital pen, *CoScribe* enables editing of paper documents as well as digital documents (PDF, PPT, etc.) on a tablet pc, synchronizing both entities. Moreover, the application allows collaborative editing and annotation in real-time. The resulting objects are sets of (collaboratively) edited, annotated, and associated documents, that can be either physical or digitalized.

As Figure 4.4 implies, no specific intended sequences for sub-processes of annotation could be identified in the performed interview. However, the system implements interaction forms of which different types of annotation activities are composed. Thus, this concrete use case enables a detection of granular steps that are contained in single phases of the annotation process, as well as the order in which these steps may be executed. In this context, the four core interaction forms *Inking* (write keywords or notes), *Clicking*, *Combining* (attach objects), and *Associating* are realized. By means of these interactions, users may conduct the annotation activities *Notation*, *Linking*, *Tagging*, and *Bookmarking*.

First of all, users may specify an area on top a document which shall be connected with an annotation. For this purpose, a vertical line may be drawn on the margin area of the

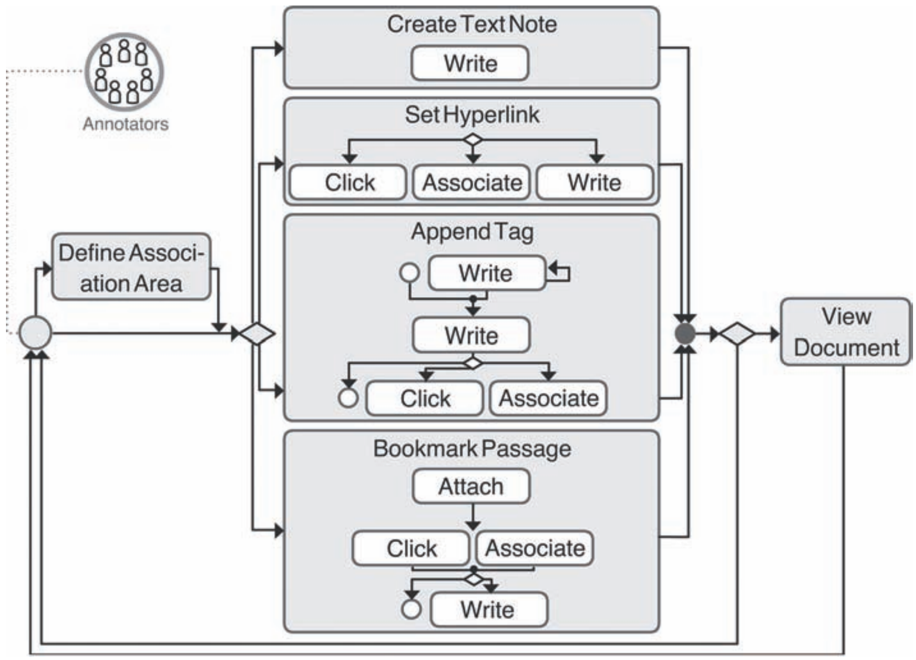


Figure 4.4: Annotation of Paper and Digital Documents based on different Interaction Types.

document, defining the range of the selected area. In addition to that, the documents provide predefined areas by default: an area on the top of the first page that refers to the entire document, and a a sidewise margin on each page that references parts of the page.

Textual notes may be created by writing the respective word or phrase on the paper or digital document. In this context, also handwriting recognition is implemented in order to foster synchronization between physical and digital copies or tagging (as explained below).

CoScribe provides bidirectional cross-media hyperlinks that connect areas within the same document, or different physical or digital documents. In general, linking can be performed in two ways: first, if documents are movable, two respective documents can be moved to a position at which the referred selection areas overlap. The factual link is generated by drawing a line between both areas. As second alternative, the user can click on the first area, holding down the digital pen for 500ms, and subsequently conduct a pen tap on the second area. Optionally, a user can write a human readable reference.

For tagging purposes, specific paper cards are provided that contain empty areas. Users can write down keywords (tags) on these areas and, in this manner, define a set of available tags. In order to assign a tag, the respective keyword is written on a selection area. Then a user may proceed in two ways: either the tag can be enclosed by drawing a circle. Here, the keyword is automatically recognized as tag from the predefined set through handwriting recognition. Second, an association activity can be performed (as with linking) between the word and the respective area on the paper card.

In order to bookmark passages of a document, users may attach post-it like stickers at an arbitrary position of a paper page, at which stickers are synchronized with the system as digital bookmarks. Optionally, these stickers can be labeled with a keyword by means of the digital pen, realizing an additional form of tagging. The single interactions of bookmarking are performed in the following order. First, the users attaches the sticker at the desired position. Second, the sticker is assigned to the digital document by means of pen association. If required, a tag can be defined by labeling the bookmark sticker.

Finally, by means of a provided visualization component, results of annotation can be explored at any time of the process. Here, a graph view shows all included documents, as well as generated links, tags, and bookmarks. Moreover, in the case of collaboration, all involved users are pictured as elements of the graph.

Annotation and Analysis of Synchronous Video Communication

At the Institute for Sports Science of Technische Universität Darmstadt, Germany, one researcher was consulted who implemented a match observation software in the scope of his Ph.D work [Lin06]. In this context, the researcher accompanied the German men's volleyball national team, whose athletes and trainers were enabled to analyze and discuss previously recorded matches in real-time and spatially separated. Using *ANVIL*, a video and audio annotation and analysis system [Anv, Kip08], conversation analyses were performed on filmed sessions of match observation, in order to evaluate and perform further development of the implemented software. As a result, annotated sets of recorded discussion on volleyball matches could be used for statistical evaluation.

The entire process can be divided into five general phases (see Figure 4.5). First, in a phase of data gathering, the conducted sessions of match observation and analysis were recorded.

Before annotating the filmed material, a theoretical framework was established, at which appropriate observation units (segments) were defined. Therefore, relevant spoken statements were chosen as selection areas. Furthermore, observation criteria were specified and inserted in the analysis software as XML coded category system. Moreover, since the employed annotation system had difficulties in processing larger video files, the recorded sessions had to be splitted into smaller units.

In the following phase, each session was categorized and transcribed. Anvil provides a timeline component that comprises multiple layers, at which each layer represents one specific category. Once a relevant statement was recognized during playback, a time interval could be defined within one layer, specifying the extent of a statement and its classification. Next, the respective transcription was entered, that is, the spoken

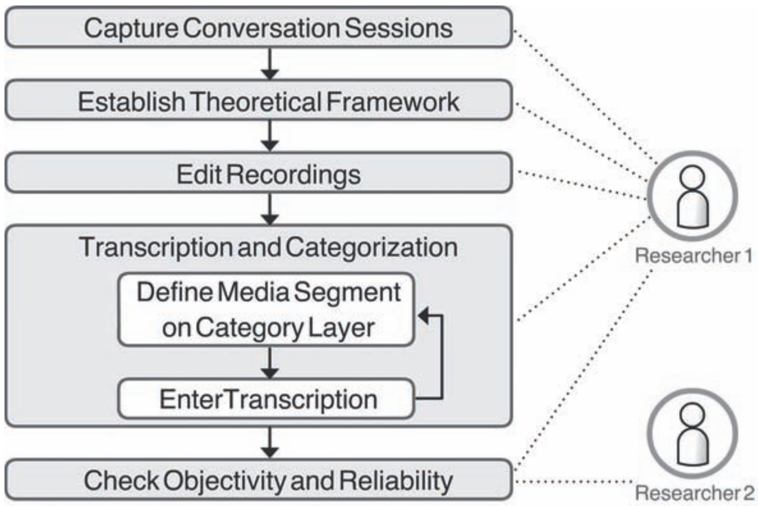


Figure 4.5: Analysis of Recorded Communication in Competitive Sports.

statement (but also further conversation features such as speaker change) was written down. This activities were repeated until all relevant statements of a conversation were transcribed and categorized.

In the last part of the analysis process, objectivity and reliability of the results were checked. Objectivity means the independence of the results of one analyst. In this context, a second researcher annotated an arbitrary sequence of six minutes. Second, in order to check reliability, which includes the trustiness of data ascertainment in particular, the primary researcher repeated the transcription and categorization phase nine months later.

4.2 Literature on Annotation Workflows

When modeling the annotation process, also aspects that derived from a survey of related literature were taken into consideration which will be briefed in this section. The presented related work concerns models of annotation processes from a theoretical view, and also field reports or proposals for “best practices” referring to a specific application domain. As will be seen in the next paragraphs, a majority of localized references belong to the field of video adaption.

4.2.1 Video Research Workflows

Focusing on the application in the learning sciences, Pea and Hoffert [PH07] present a model that describes the general procedures of video research that is based on video annotation and analysis. Figure 4.6 shows that a project usually starts with planning (storyboard, cameras, additional data capture, etc.) and video preproduction activities. After recording video material, the obtained files are encoded to appropriate file formats and stored, which includes sorting of files and labeling, e.g., for retrieval purposes. The videos are then segmented by defining event boundaries or time markers. In the following, researchers continue with selecting relevant segments, transcribing observed facts, categorizing segments, and analyzing data in an iterative manner. Additionally, these activities are always accompanied by searching, tracking, and browsing. In general, it is pointed out that there are interdependencies between acts of selecting segments, transcribing, categorizing, searching, and browsing, since they are alternately executed in the course of recursive passages. The next workflow phases include presenting and sharing analysis results. Here, sharing refers either to discussion on a recent video analysis project, or publishing in a journal or DVD for example, at which the results are commented by a larger community. Finally, Pea and Hoffert describe that analysis results may also have an impact on following video research projects and workflows.

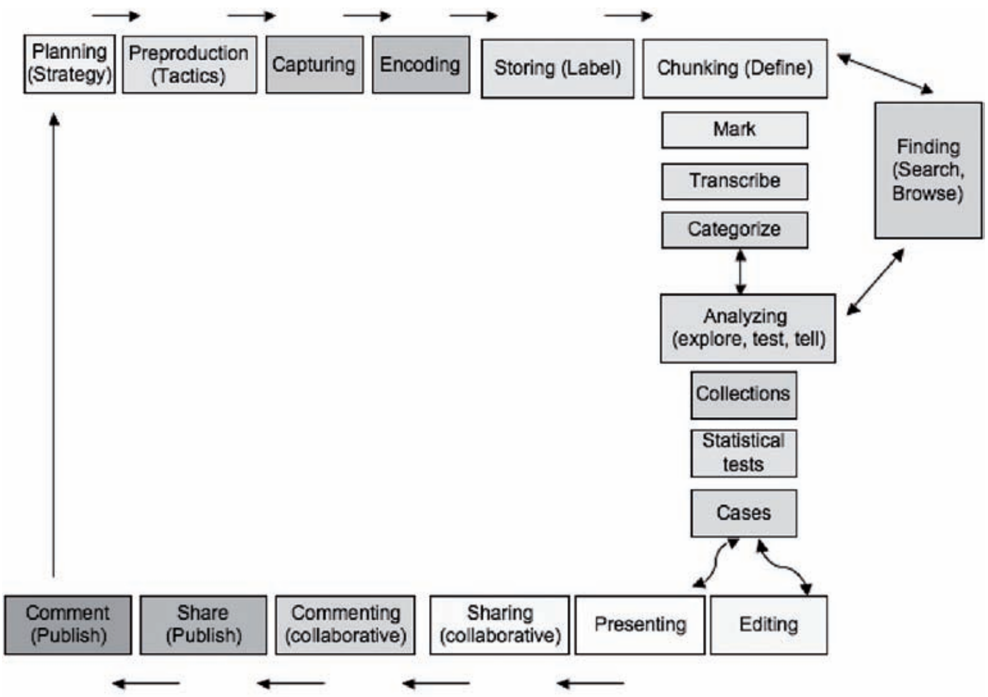


Figure 4.6: Phases of the Video Research Workflow. [PH07]

Comparable to the video annotation and analysis procedures conducted at the IPN Kiel (see Section 4.1.2), Mikova and Janik [MJ06] report on the *CPV Video Study Physics*, which was conducted in order to analyze filmed lessons of physical education. Here, they also propose a methodological procedure for such kind of research work. Basically, they divided the analysis into three phases. First, preliminary interviews with the teachers were held, and afterwards the lessons were filmed with multiple cameras. The second phase included data preparation. Here, the interview data was transfered to SPSS, and recordings were encoded to suitable files. Then, the videos were segmented into fix units of 10 seconds, transcribed, and categorized with respect to a predefined category system. By analogy with the IPN study, these activities started after a previous training of researchers. In the third phase, the gathered information was statistically analyzed by means of SPSS. Finally, selected videos or cutouts were published as examples of "good teaching".

Harrison and Baecker [HB92] proposed requirements for video annotation and analysis systems, particularly based on a performed task analysis at which users of different annotation systems were interviewed. As a result of the tasks analysis, the two process-related points *annotation* and *detailed analysis* were derived. According to this, first annotation is performed which refers to note taking, including the personal capturing of relevant data in real-time. Thereafter, the captured data is analyzed, at which segments of video are transcribed in several iterative passes. Moreover, statistical analysis tools are applied, and results are summarized by means of specific visualizations.

In order to support the video annotation and analysis workflow, Hagedorn et al. [HHK08] specify the annotation and analysis workflow based on conducted expert interviews and a State-of-the-Art analysis. As described in their paper, the workflow steps generally proceed as follows: After collecting the required video material, segments are defined as areas-to-annotate. Then, training units of annotation are performed including checks for reliability. In the next steps, the annotation is performed, followed by repeated reliability checks and discussion of potential disagreements. In the last phase, data analysis is executed. Here, it is important to point out that the approach of Hagedorn et al. is not comparable to the concepts presented in this thesis, since an actual support of sequences and transitions between workflow steps is not explicitly emphasized.

4.2.2 Qualitative Analysis of Multimedia Documents

As described in Section 3.2.2, *ATLAS.ti* is a system that supports qualitative analyses of text, image, audio, and video documents. Within the respective user manual made available at [Atl], a general procedure for the analysis process is proposed (see Figure 4.7). In this process description, first a project is created (*Hermeneutic Unit*), including configuration of required project properties and the media documents (*Primary Documents*) that are to be analyzed. Next comes the actual annotation phase, in which segments of the incorporated media documents are defined, assigned to predefined categorical keywords (*codes*), and enriched by textual notes (*memos*). In the following step, the media segments are compared based on attached annotations and, if required, further media files are assigned to the project. This comparison often leads to a (re-) organization of annotation data, for instance, by defining groups and hierarchies. After that, analysts may build networks, specifying links or relations between existing information entities. In the last phase of the analysis process, results are either published, e.g., as a Web document, or exported in order to perform further analysis with a statistical software.

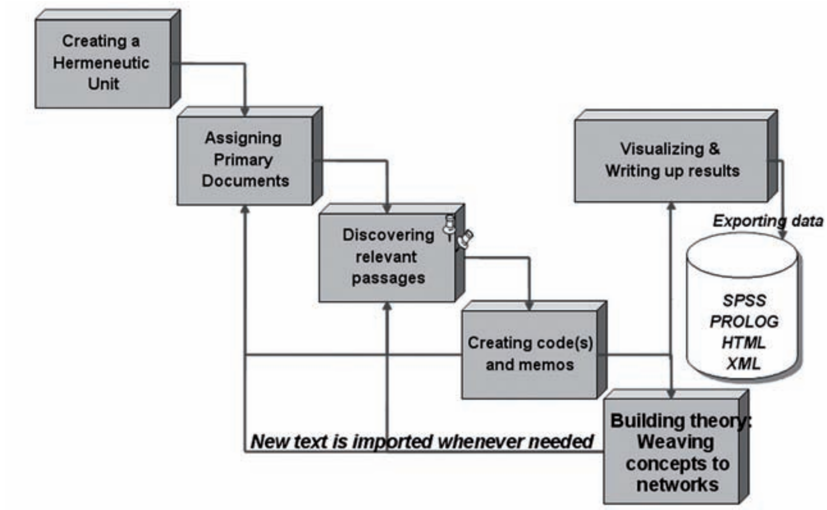


Figure 4.7: The Qualitative Media Analysis Workflow proposed by ATLAS.ti. [Atl]

4.2.3 Media Processing and Media Content Analysis

Both media processing and content analysis are to be seen as strong related to annotation, since additional information is attached that arises from the automatic machining of multimedia content. Thus, the terms *processing*, *content analysis*, and *annotation* are frequently used in synonymous manner. Seiler et al. [SHJ$^+$08] point out that processing or content analysis workflows mostly follow a sequential structure. As pictured in Figure 4.8, the process is structured into training and analysis, which both comprise almost the same substeps. First of all, in order to create required (use case dependent) data, preliminary tasks are performed, such as retrieving media files, decoding video frames, or decompressing audio. Then preprocessing is executed, for instance, transformation or filtering. After automatically detecting (mostly low-level) features, the processed data is classified. Finally, the obtained set of data can be transfered to further postprocessing tasks.

Another example is described by Tsai and Hung with respect to automatic image annotation [TH08]. The paper reviews fifty systems that use machine learning techniques to annotate image for retrieval purposes. In this context, a general procedure of the automatized annotation process is presented. First, segmentation of image documents is executed, at which the image is divided according to blocks or detected regions. Then, visual features are extracted, e.g., colors, texture, shapes, or spatial relationships. In the next step, the gathered information is annotated. That is, detected low-level features are attached to the image with regard to the partial areas defined by segmentation.

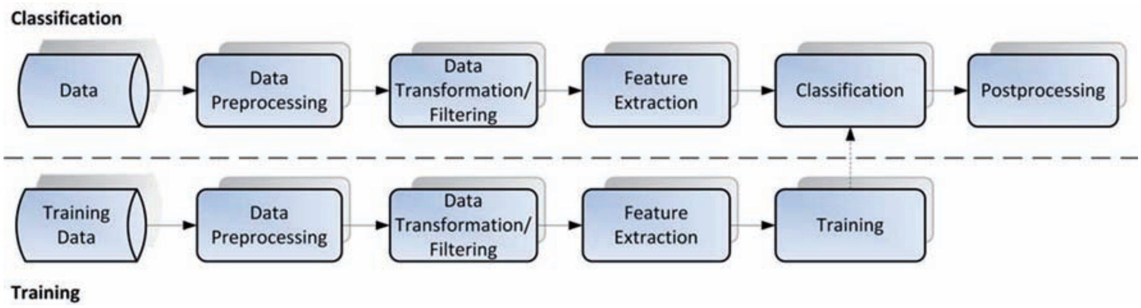


Figure 4.8: The Multimedia Content Analysis Workflow. [SHJ+08].

Afterwards, in a phase of post-processing, visual features are assigned to semantics. Here, features are recognized or classified according to predefined categories.

4.2.4 Text Annotation for Natural Language Processing

Knowtator is an application to generate documents of annotated text that can be evaluated by Natural Language Processing (NLP) systems [Ogr06]. The generated documents comprise arbitrary text and structured annotations that arise from a predefined annotation scheme. Within the *Knowtator* documentation pages [Kno], an exemplary annotation workflow is described which has been set up by the developers in lab situations. Here, the first step is to define an annotation scheme, determining classes and instances which are assigned to text passages during annotation, as well as instances that represent the annotators that will work in the project. After that, the annotation task (i.e., the annotation scheme) is distributed to the participating users. Then, annotation is performed, at which parts of the text are assigned to one or more annotations of the employed scheme. Once multiple annotators have accomplished their individual annotation tasks, results can be connected by merging the different documents, which is explicitly featured by the system. Potentially, this leads to a re-editing of the defined annotation scheme. In addition to that, automatically reliability checks can be executed (which also may lead to a modification of data). Finally, the entire document is exported as XML file in order to be processed by NLP applications.

4.3 Conclusions: Empirical Study and Literature Survey

A feature-oriented analysis of annotation systems (Chapter 3) primarily revealed tools, functionalities, and approaches that are applied in order to process certain annotation tasks, as well as forms of data that result from these features and are transfered between different tools and users. In addition to that, a conducted empirical study and a survey of related literature aimed at mapping these functional units to task-related phases within the annotation process, as well as revealing relations between these phases with regard to operational procedures and execution orders. In the following, general conclusions that result from the studies on annotation practices are drawn.

In general, phases or sub-processes within the entire annotation process can be detected, which are similar in the different described cases and thus can be mapped mutually. First of all, preliminary activities can be identified that refer to the scheduling or planning of the annotation process and the configuration of the applied system. Moreover, further workflow sub-processes can be derived that refer to the actual tasks of annotation. These include the definition of parts that are going to be annotated on the one hand, and acts of information attachment on the other hand. Some of the investigated workflows (or models) make clear that these acts of annotation are often accompanied by data verification, that is, users constantly search and browse own annotations or results from co-annotators. This can lead to the revision and modification of results and, consequently, to jumps and iterative loops within the process. Additionally, it is also exploited in order to train annotators or, in the case of machine-learning based processing, also systems. Finally, the resulting documents of annotated media files are exported for further processing or published in diverse distribution formats. Thus, the results of an annotation process may have impact on following projects.

The different annotation workflows described in the previous sections show that it is not possible to declare a “best practice ”of annotation. Although commonalities could be demonstrated that allow a summarization of activities into process phases, the execution order of sub-processes may differ strongly to some extent. For instance, the exemplified workflows show that automatized approaches mostly evince pipeline-like structures, while workflows in which user interaction is required provide more networked task interconnectedness. Thus, even though different processes comprise identical tasks, the positions within the entire workflow may differ. A representative example is the segmentation task, which is mostly executed during the factual annotation phase, but is also displaced to preliminary activities in some cases. This concludes that the establishment of a general process model for annotation requires a suitable degree of abstraction. Second, it justifies the idea of permitting users (or administrator) to individually specify the annotation process by means of workflow management approaches, which constitutes one of the main contributions of this thesis.

The following chapter summarizes the aspects and results described in this section as well as in the scope of a systems analysis conducted in Chapter 3. Here, a *Generic Process Model of Multimedia Annotation* will be presented, which has been developed in the course of this work.

Chapter 5

A Generic Process Model of Multimedia Annotation

In order to develop a workflow-based approach to improve the interactive operation of multimedia annotation systems, a fundamental step is to specify the annotation process, establishing knowledge about how persons actually work with respective systems. According to these viewpoints, a *Process Model of Digital Multimedia Annotation* will be presented.

The model elucidates the concrete process taking place, splitting it up into phases and sub-processes. Furthermore, a survey of (i) possible tasks that have to be accomplished, (ii) types of annotation data and other information involved in these tasks, and (iii) different tools, functionalities, and algorithms that are used to handle tasks and data, is given. At the same time, these different aspects are assigned to the identified phases and sub-processes annotation.

Due to the wide range of annotation with respect to its different appearances, functions, purposes, application domains, and media formats (and consequently its different workflows), the process model requires sufficient degree of abstraction to capture diverse facets of annotation in an appropriate way. For this reason, the described model has to be regarded as a *Generic* process model.

This chapter is organized as follows. First, the underlying methodology with respect to the model-building work is described in Section 5.1. The central issue of this chapter, a the developed generic process model is presented in Section 5.2. In Section 5.3, the validity and applicability of the established model will be discussed. Finally, the single results comprised in the model are summarized the last part of this chapter.

5.1 Methodology of Process Modeling

For process modeling purposes, the preliminary examination of annotation systems and practises were brought together. In the scope of the analysis of different types of multimedia annotation systems conducted in Chapter 4, identified tools, functionalities, approaches, and types of generated data were assigned to the respective classes of tasks and activities which were derived from an inductively developed category system(cf. Section 3.4). Additionally, in the context of the examination of annotation practices, different workflows of multimedia annotation were derived by conducting expert interviews and observations, and by surveying published workflow models, field reports, and execution proposals. By means of a the summarization and generalization of these workflows, general phases, sub-processes, and included tasks, as well as sequential relations between these items were abstracted (cf. Section 4.3).

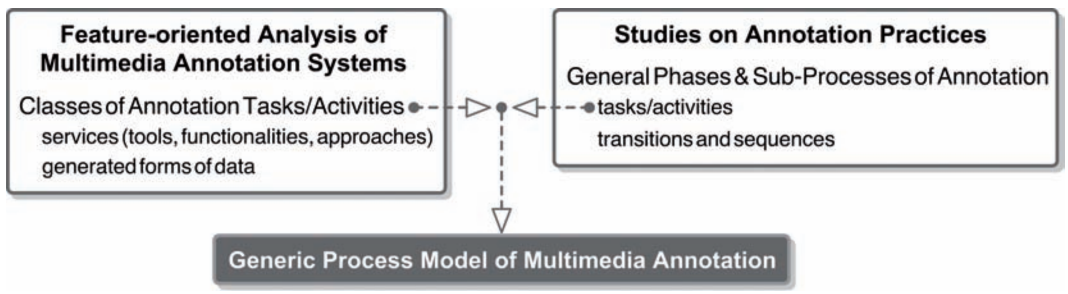


Figure 5.1: Methodology applied for Process Modeling.

As illustrated in Figure 5.1, the findings of both preliminary investigations were combined in order to construct the generic process model described in this chapter. Here, classes of tasks and activities obtained from the system analysis were assigned to tasks within phases and sub-processes of annotation identified in the conducted studies. In doing so, also indetified features and sequences were combined, so that the entirety of all imposed information could be summarized within a common model. In the following section, the resulting *Generic Process Model of Multimedia Annotation* will be presented.

5.2 A Generic Process Model of Multimedia Annotation

In this section, a *Generic Process Model of Multimedia Annotation* is presented which describes how people effectively work with multimedia annotation environments. In this context, knowledge is provided about (i) the general phases, sub-processes, and tasks of annotation, (ii) the incorporated tools, functionalities, and approaches applied to execute these tasks, and (iii) the forms of annotation data that are generated within workflow steps or passed through different tools or functionalities. These results constitute a basis for the conceptual design of a process-based annotation framework, which will be illustrated in Chapter 6.

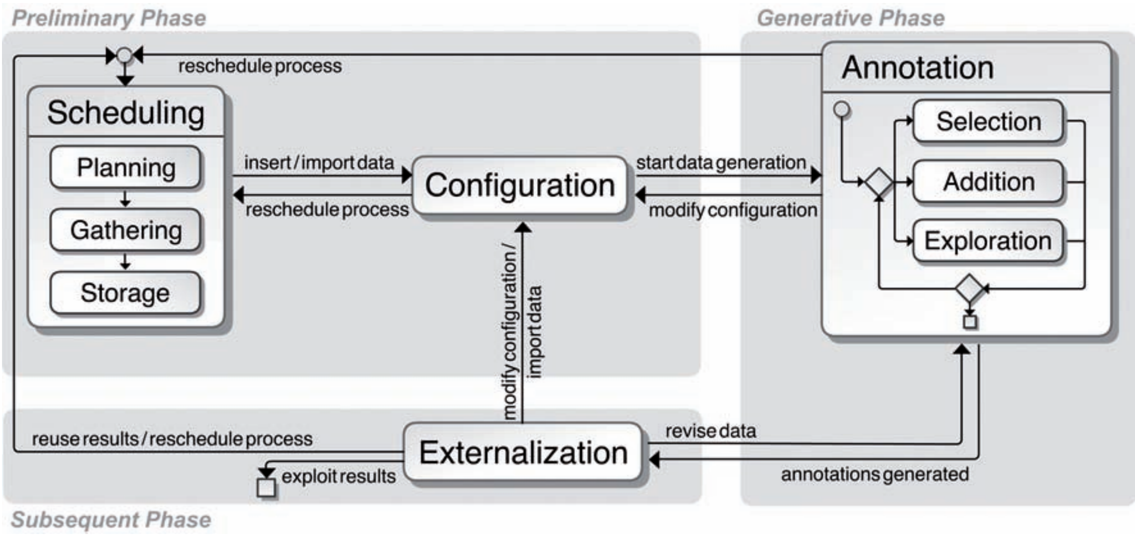


Figure 5.2: Schematic overview of the Generic Process Model of Multimedia Annotation including phases, sub-processes, and sequential interdependencies.

As shown in Figure 5.2, the established process model not only refers to the actual activities of annotation, but also considers processes which take place before and after annotation, as well as without usage of the annotation system. Accordingly, the entire annotation process can be structured into the three superior phases *Preliminary*, *Generative*, and *Subsequent*. The initial Preliminary Phase comprises all activities that might need to be performed before users actually start annotating. Here, the sub-processes of *Scheduling* and *Configuration* are included, which refer to planning, defining strategies, gathering data, or preparation of the annotation environment. In addition to that, sub-processes can be detected that are to be assigned to the Generative Phase, in which *Annotation* is effectively done. Here, Annotation is subdivided into the partially processes *Selection* of validity areas and *Addition* of the supplementary data, which are accompanied by acts of *Exploration*. Furthermore, *Externalization* procedures refer to the further processing of already annotated data, passing into a Subsequent Phase that is not conducted by means of the annotation system and might imply switching to another project.

In the following, the described sub-processes *Scheduling*, *Configuration*, *Annotation*, and *Externalization* are explained. In order to provide a better understanding, examples of cognitions acquired in the scope of the feature-oriented system analysis (Chapter 3), the empirical study (Section 4.1), and the literature survey (Section 4.2) are given for each aspect.

5.2.1 Scheduling

In particular, the conducted empirical study showed that preparatory measures on a conceptual and organizational level are essential in some use cases and might be crucial to the quality of acquired data. This applies especially to annotation processes conducted in investigational settings. *Scheduling*-related tasks regard to the design of the annotation process in general, as well as to *gathering* and *storage* of media files that are going to be annotated. Figure 5.3 gives an overview of the procedures contained in this process item.

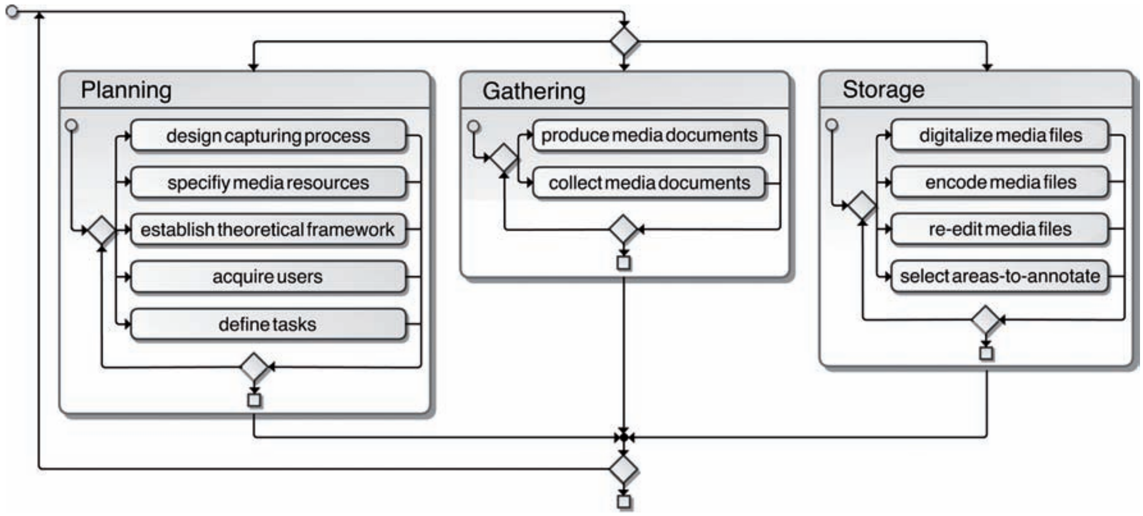


Figure 5.3: Procedures of Scheduling Sub-Processes.

Planning

At the beginning of a project, several decisions have to be made. If the incorporated media files are produced by the project participants, the capturing processes need to be scheduled by defining, for instance in the case of video recordings, the number of involved cameras, if lighting is required, or whether a storyboard needs to be created [PH07, Rat03]. In this context, it must be considered if there is any additional data (like sensor data, eye-tracking data, or an additional audio track) that has to be captured in parallel with the recordings [BR04, HHK08]. On the other side, resources such as multimedia databases must be determined, from which the subsequently processed media files are extracted.

From a methodical view, a theoretical framework might need to be built up. As shown by the described video studies on lesson recordings and sport match observations (Section 4.1.2 and 4.2.1), such a framework implies the identification of research areas, research questions and hypotheses [Lin06, MJ06, SPK05]. Additionally, the definition of categorization schemes is often required, which potentially determines the structure, terminology, or domain of descriptive metadata that will be annotated. In this context, Stamou et al. [SOPS06] found out that choosing the right vocabulary is one key decision in an annotation process. In the case of qualitative research, such schemes may be either developed deductively based on a theory, or inductively based on the given multimedia material [BD06, MJ06, SPK05].

The constitution of a theoretical framework, in combination with the storage of media files explained below, allow the specification of concrete annotation tasks that will form the information generation process. In collaborative use cases, these tasks can be assigned to users and/or groups participating in the annotation process. Accordingly, planning might also comprise the acquisition of users as well as task distribution.

Gathering

In a next step, the media files and additional data need to be recorded or collected. Examples of media capturing are filming video, recording audio, shooting pictures, writing a text, or designing a webpage [Bul03]. If required files already exist, this data can be gathered from specific databases or storage media [HHK08, MJ06, PH07].

Storage

Depending on the format of the captured or collected data, the media documents might need to be encoded or digitalized to suitable files [MJ06, PH07]. Then, the data is organized and stored [PH07]. In this context, especially the interview held at the Institute for Sports Science at the Technische Universität Darmstadt, Germany (Section 4.1.2) revealed that this can lead to a re-editing of the media files to more granulated units (cf. [HHK08, SFZ06]). According to Bulterman [Bul03], the original raw media is usually not saved. In some cases, for example in the scope of the IPN Video study (Section 4.1.2) or the operational design of the *EVA* system (Section 3.2.5), this sub-process also may include the definition of areas-to-annotate, for instance, by performing video or image segmentation.

5.2.2 Configuration

According to conceptual determinations made during Scheduling, specific properties must be transfered into a machine-understandable form in the course of administrative entries [Bul03]. The conducted analysis of annotation systems showed that these properties generally refer to the customization and management of users and data involved in the process on the one hand, and to the configuration and storage of general preferences on the other hand (cf. Section 3.3 and [BR04]). The latter aspects may include the customization of the user interface elements with respect to their visual appearance, or the management of tools that may be integrated in a superior environment.

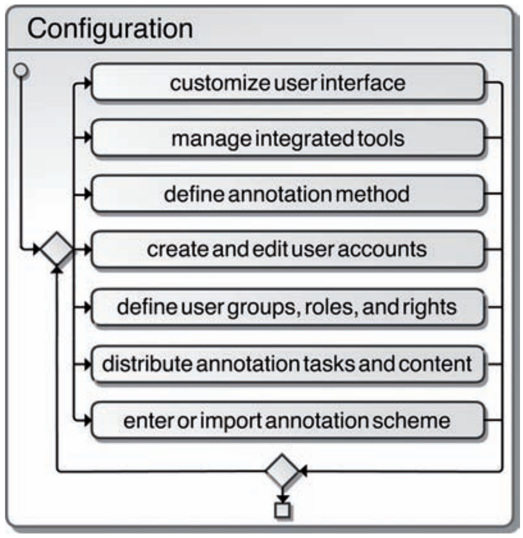


Figure 5.4: Administrative procedures of Annotation.

For instance, *M-OntoMat-Annotizer* provides plug-in management, and *Music Annotator* allows the integration of different feature extraction tools. Another frequently provided option is the configuration of the annotation method (*ELAN*).

Further preferences are the involved users and user groups. Here, the general tasks of a project can be assigned to predefined groups, or a "group administrator" might distribute the annotation tasks among the individual users [SGRH07, VSN05]. A representative example is *Music Annotator*. Here, users can select tasks from a provided to-do list. Another type of task distribution is implemented in the *EVA* system, at which so-called workloads (sets of files to annotate) are assigned to different annotators. In this context, users and groups can be associated with specific roles that particularly include access rights and restrictions, such as realized by *Vannotea*. If the annotation process is connected to a specific annotation scheme, this scheme must be integrated by either generating and editing, or importing it from an external resource. Examples for such schemes are (controlled) vocabularies, categorization systems, sets of low-level features, or more structured semantic infrastructures and ontologies.

Since annotation processes are iterative and contain loops and re-entries to previous process states [PH07, SFZ06], predefined configurations of the used environment often need to be modified during a running annotation process. Associated with it, also rescheduling activities must be conducted.

5.2.3 Annotation

The next sub-processes, *Selection*, *Addition*, and *Exploration*, can be regarded as one operative unit. The conducted expert interviews revealed that no specific operative sequences can be identified referring to these "real" acts of *Annotation*. Pea and Hoffert

[PH07] note on video analysis processes that activities of decomposition (segmentation, categorization, and transcription) and re-composition (rating, interpretation, reflection, comparison, and collocation) are closely interrelated. They depict video annotation as a complex process that contains circular and recursive loops, in which the analyst alternately marks, transcribes and categorizes, analyzes and reflects, and needs to conduct searches. Marshall and Ruotolo [MR02] performed a field study with respect of the annotation of digital libraries, reporting that acts of searching, reading and annotating are performed at the same time and can be done together with other activities, e.g., working with colleagues. In addition to these aspects, it has to be mentioned that selective and additive interactions are often performed on user interface components that are primarily designed for data visualization and navigation purposes. For example, reporting on video annotation, Kipp describes that segmentation and attachment of additional information is usually performed on specific timeline components [Kip08]. Hence, Selection, Addition, and Exploration, as higher-level categories, have to be considered related to each other, basically not disposed in fixed executive sequences. Nevertheless, it has to be mentioned that it is yet possible to arrange sub-processes in sequence. This particularly applies to Selection and Addition, what has been shown by some of the examined annotation use cases (cf. Section 4.1.2). Figure 5.5 pictures the operational interrelations between these sub-processes. Furthermore, it illustrates respective tasks for each sub-process, which will be described in the following.

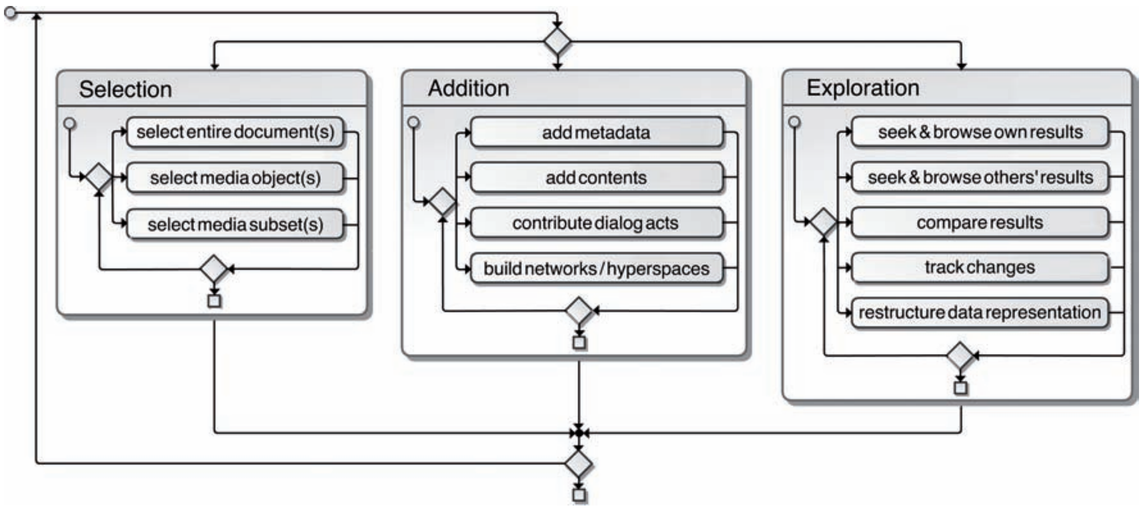


Figure 5.5: Three main Sub-Processes of the Annotation Phase.

5.2.4 Annotation: Selection

Annotators need to mark specific contents of interest at which annotations shall refer to, i.e., digital contents need first to be declared as “annotate-able”. As illustrated in Section 2.1.4, there are different types of areas that can be connected to annotations, defining their validity and context within the media document. Hence, selective activities are closely connected with the field of *anchoring*. In this context, the simplest task is marking a whole multimedia document such as a website. A second variant is the selection of elements contained within a document. Assuming that a considered web document consists of a text, various graphics, and a video, an annotator is able to mark one or more of these elements as encapsulated units, and subsequently annotate them with different information [Fin05, Rei06]. As a third variant, these single media objects can be again subdivided into content subsets or segments. For example, annotations may be associated either to an entire text, or to a paragraph within the text [CDTT04]. Cadiz et al. [CGG00] describe such kind of annotations as *in-context-annotations*. In the scope of collaborative use cases in which groups of users generate annotations on shared data sets, especially the consideration of the systems *ELAN* and *Vannotea* showed that selective activities on such a fine-grained level can support co-annotators by providing possibilities of "pointing" on content they want to modify or discuss on, e.g., by means of highlighting or drawing features (see also [Fin05, Kip08]).

How exactly subsets or segments are defined, depends on the media format of the original content and its specific properties, as well as on the purpose of annotation. In general, annotators may exploit manual, semi-automatic, or automatic techniques. Here, the concrete available selection methods or algorithms determine the required degree of human activity. An example for *manual segmentation* can be localized in the field of web pages, at which image regions can be defined by means of *image maps*. Here, different parts of an embedded graphic file can be marked by specifying pixel-based coordinates which enclose the required image area and can be individually associated with hyperlinks. A case of *semi-automatic selection* was found in the examined university courses on hypermedia design described in Section 4.1.2. Here, an early applied annotation system allowed students to define so-called sensitive regions on videos by means of a keyframe-based method. This method enables users to define wireframes that enclose and track an object of interest at different points in (playback) time, at which tracking is realized by automatic interpolation of subsequent wireframes [Fin05]. *Automatic segmentation* approaches execute selective tasks without the inclusion of human interaction, intending saving of time. In the field of video, common approaches are object or shot detection, scene-based event logging, or object of focus detection [BCQ+04, BDBCP04, SW05]. With regard to automatic image segmentation, a survey of recent methods is given by McGuinness and O'Connor [MO10]. Further examples have been illustrated in Section 2.1.4, providing an overview of recent research on digital media annotation.

In general, especially the conducted system analysis revealed that the following types of data arise from area selection: If an entire document or file is to be annotated marked, it can be referenced by a URL or URI, or a path that points to the local file system. Within a multimedia container, one or more embedded media objects can be addressed. Here, numerical spatial information is frequently obtained, for example, X-

Pointers or HTML anchors. In the case of selecting an area within a media object, also numerical metadata is attached in order to point on this area. Here, the concrete type depends on whether the annotated media object is to be regarded as continuous or discrete (cf. Section 2.1.4). Accordingly, metadata can represent spatial information (e.g. coordinates), temporal extents (time intervals, time marks, etc.), or spatio-temporal combinations (e.g. wireframes for object tracking).

Media subsets or segments also may be artifacts of collaborative work. As showed by the university courses on hypermedia design (Section 4.1.2), the segmentation tasks can also be distributed among different collaborating groups [SFZ06]. Here, distribution criteria can be different characteristics, for example, group A chunks the video according to a certain characteristic 1, group B seeks for characteristic 2, and so on. Thus, if classification systems, vocabularies, ontologies, etc. are used, these may differ for each participating user or group.

5.2.5 Annotation: Addition

After marking the relevant documents, document subsets, or object segments, annotators continue adding annotations as additional information to these elements. Here, the type of information that is going to be annotated determinates the tools, approaches, or algorithms that are required. These tools in turn determine the user's degree of active involvement and patterns of interaction. According to the classification of the different appearances of annotations (*metadata*, *content*, *dialog act*, and *hyperspace element*) presented in Section 2.1.3, various tasks of inserting additional information are to be distinguished (cf. Figure 5.6).

Addition of Metadata

A relevant objective of annotation is the classification of given information resources or parts of it, for instance, in order to facilitate archiving and later retrieval [BBN05] (cf. also Section 2.1.3). In this context, Steimle et al. [SBM08] noticed that annotators always use specific categorization schemes (even if it is not explicitly demanded). In order to categorize contents, users may insert some kind of metadata which may range from simple content features to complex semantic information. One example is tagging, which Baecker et al. [BFW07] regard as a useful method to collaboratively organize amounts of multimedia information. While tagging comprises a free choice of terms that are used for classification, more structured guidelines are provided by (controlled) vocabularies or categorization schemes. Here, a specific taxonomy is predefined which forms the basis for category-based annotation. In this context, the given taxonomy might have been developed in the scope of investigational work. In this sense, video analysts at the IPN Kiel use *Videograph* to assign one or more predefined categories to a current time interval via keyboard shortcuts [Rim05, SPK05]. A more

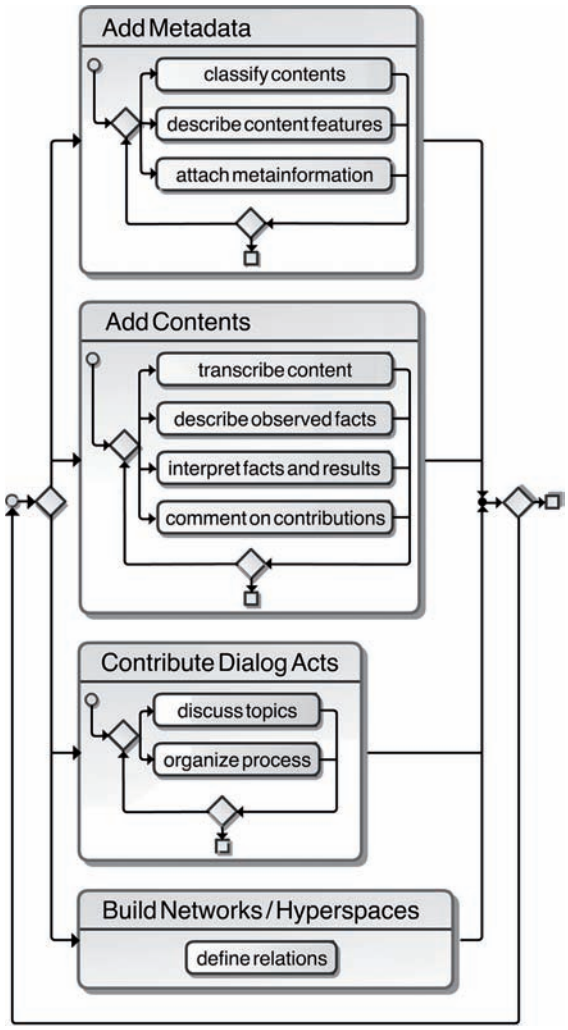


Figure 5.6: Different Forms of Information Addition.

complex approach of content categorization is the addition of semantic information, aiming at enhancing human-interpretable data with well-structured meaning in order to create computer-interpretable descriptions [BLHL01]. Since annotated semantics comprise multiple information such as classes, instances, properties, or relations (cf. the *M-Ontomat-Annotizer* system), more human interaction is required with respect to information assignment, modification of the data representation, and information re-editing [PH07].

In addition to content classification, a further task in the context of metadata addition is content description, that is, description of facts, properties, or events that can be observed or are contained in the media document. Unlike content description by content-type annotations (see next section), metadata-based annotations are to be regarded as low-level features. For instance, as supported by *M-OntoMat-Annotizer*, these can be MPEG-7 visual descriptors such as color. Moreover, patterns of images or “moving-images” may be described [BBN05]. As shown by *Music Annotator*, low-level description (like “beats per minute”) might be provided. Hence, not only visual but also aural features may be of interest (cf. [VSN05]). A further example is keyword extraction, which can be applied to textual documents, abstracting relevant terms.

Another form of annotation tasks in the scope of metadata attachment is giving additional metainformation about the annotated media object, defined segments, or already existing annotations. Such kind of tasks are usually regarded as the traditional forms of metadata generation. Here, as supported by *EXMARaLDA*, *MADCOW*, or *Vannotea*, metainformation can be attached such as annotation or document type, author, title, creation or modification date, location, or context.

Bookmarking is a further way of applying metadata-typed annotations [CGG00, Mar98]. This enables users to mark and store relevant contents or portions for later inspection. Bookmarking can also have the objective of supporting collaborative work. In this case, users are enabled to see contents that seem to be relevant to other members of their group [KD09]. *EMargo* for example (see Section 4.1.2), allows attaching of “flags” which serve as shared bookmarks with respect to paragraphs within a digital script [SGRH07]. By means of *CoScribe* (see Section 4.1.2), digital paper bookmarks are first digitalized and fed into the system and afterwards presented by a collaborative visualization [SBM08]. Analogous to the *Selection* task, bookmarks are represented by defining the context or position of the content of interest. For this purpose, a URL, URI or path description can be attached, but also inter-document and inter-media entities like segments, X-Pointers, etc.

While the illustrated examples refer to manual metadata generation and attachment, the analysis of annotation systems revealed that also semi-automatic or fully automatic approaches are incorporated for this kind of annotation tasks. For instance, *Music Annotator* employs automatic extraction of aural features or ID3 descriptors. To take another example, Bertini et al. [BBN05] presented a system that supports automatic extraction of video features in order to permit automatic summarization of sport videos. The latter example shows that automatic approaches may also assist to other relevant activities such as the export, summarization, or publishing of results.

Addition of Contents

In certain cases, users may need describe observed facts, for instance, behavior and events of a video, objects within an image, sequences of an audio file, etc. In analysis cases, a task can be the transcription of verbal and nonverbal communication, which is often used in the context of communication or interaction analyses [MJ06]. In order to explain, elucidate, interpret, or comment on the given contents, users need to give descriptions in a more free way than assigning metadata [AF08b, MJ06, SPK05]. For this purpose, they can use tools allowing them to enter free textual annotations. Within the university courses attended by *eMargo* (Section 4.1.2), students are allowed to annotate provided textual lesson scripts with questions or remarks on the course topics. These contributions are visible to fellow students and teachers [SGRH07]. An example for automatic approaches related to this field of tasks is speech-to-text, at which recorded speech is automatically transcribed into textual form.

For the same purposes, also other types of media formats can be annotated, ranging from textual annotations to image, audio, and video annotations [AF08b]. For example, as described in Section 2.1.3, the video platform *Youtube* [You] enables not only textual commentary but also the aggregation of videos (called “video-reply”). Another example is the analyzed system *AnnoCryst*, which allows the the annotation of 3D-Models with arbitrary multimedia content via URL or path specification. In the inspected video analysis use cases, the annotation phase includes interpretation, rating, and reflecting. These activities can be performed either qualitatively, e.g. in discussions, or quantitatively, by means of statistic methods provided by specialized software [FHM⁺01, PH07]. Especially the usage of different media encoding is to be regarded as instrument for elucidation or further explanation of existing content. Thus, each medium has specific advantages and disadvantages with respect to the explanation of certain types of facts or information in general. For example, video-based media is suitable for the representation of dynamic processes [Fin05], while textual media is an instrument to express thoughts, feelings, etc. Consequently, annotation of multimedia content can be means of explanation from different viewpoints.

Like the selection task, the generation of annotations can be divided and distributed to different users and groups. For example, in the courses performed with *eMargo*, students are divided into groups which are assigned to specific tasks. For each task, one student obtains the role “group administrator”alternately. This user is responsible for the coordination of the collaboratively elaborated tasks, as well as for unblocking of a final version [SGRH07]. Thus, any annotator has got access to the group’s selections and annotations and is allowed to conduct modifications. In that case, annotated information becomes a shared contribution [AF08b, CDTT04, FBT⁺03, ZBM06].

Contribution of Dialog Acts

Some of the analyzed annotation systems provide the possibility to attach annotations on already existing annotations. In the case of textual contributions, this is a means of establishing communication around annotated objects between a group of users that have specific access rights. Additionally, specific tools such as chat or discussion boards may be applied. Communicational contributions constitute an essential kind of shared annotation with respect to collaboration. They enable co-annotators to discuss content-related or organizational matters.

When users work separately, they need to discuss their annotations, conclusions, and the analysis process with other participants (in synchronous and/or asynchronous manner) [BR04]. Current applications usually realize group communication by providing textual comments similar to web forums. Within a *WebDIVER* project, users at the KRMC are able to respond already submitted textual interpretations or comments (cf. Section 4.1.2). In doing so, a discourse is realized making use of a hierarchical construction of annotations. In the performed courses on hypermedia design (Section 4.1.2), one of the first group tasks was to discuss the video segments that had to be specified, before actually feeding the used annotation system with this information [SFZ06]. Particularly in the context of consensual approaches applied in content analysis, discussion is a means of agreement and consistency of different annotators results. Discussion often leads to a return to previous steps of the annotation process. In the case of video analyses performed by the IPN, several analysts first classify the same video segments independently and afterwards compare the individual results with their co-analysts. In case of disagreement, the data is reviewed and modified as a result of held discussions [SPK05]. In addition to that, the interviewed experts at the IPN report on a training phase that is conducted before the actual video analysis on new video material [SPK05]. This phase aims to develop basic analytic skills [SFZ06], perform checks for objectivity and reliability, applying different annotator agreement measures [FHM$^+$01, Lin06, MJ06, SPK05], and to validate the deployed category system [SPK05]. As a consequence, these checks lead to a return to processes of *planning* and *configuration* [MJ06, SPK05]. In the end, the final results of the annotation project arise from iterative loops through the process, in which the data is continually modified and adjusted.

Organization and coordination of collaborative activities are essential in the context of co-writing or co-authoring with respect to granulated information exchanges between collaborators creating a shared document [BFW07, CGG00, ZBM06]. For this purpose, authors can generate different forms of contributions: *To-do items* that determine tasks, *summaries* of edits which were performed by co-annotators, *discussion* about the content that can be subdivided into questions and general comments, and *comments on exiting comments* [ZBM06]. In the online-courses attended by *eMargo*, student groups have to distribute the set of tasks and determine a responsible group administrator for each single task by themselves. For this purpose, they use an integrated commentary tool [SGRH07] (cf. also Section 4.1.2).

Building Networks and Hyperspaces

As already stated, annotations enable the creation of new relationships in the form of link source(s) and destination(s), connecting annotations with existing contents [AF08b, CDTT04, Fin05]. In this case, the media content and its annotations establish a inter-connecting hyperstructure [ABDF07]. This enables recipients to obtain multiple views and consequently new perspectives on existing information (cf. [MB04, SFZ06, VSN05]). In addition to that, further navigation and reception options are provided including alternative paths that can be pursued. Moreover, enhanced search functionalities are revealed [ABDF07, Fin05].

In order to create hyperstructures, annotators need to operate specific net building services (tools, functionalities, etc.). The concrete form of such services, as well as the associated way of interacting with the system, depends on the specific characteristics of the contents that are to be linked [CCG02]. For example, a website is referenced differently than parts of it, or, a moving video object has to be linked other than an object within an image, since object motion is additionally tied to temporal information [CCG02, Fin05]. Beyond that, it has to be considered that also annotations can be destination of a hyperlink. Hence, annotations themselves can be regarded as "existing content" with specific properties.

5.2.6 Annotation: Exploration

The tasks of *Selection* and *Addition* always go along with searching, browsing, and reception activities [MR02, PH07]. First of all, surveying ones own data is required to properly perform digital annotation [PH07]. Here, users are allowed to review and evaluate results, and to return to previous process steps or phases in order to modify data. As stated by Marshall and Ruotolo [MR02], acts of searching, reading and annotating can also be done together activities such as working with colleagues. Consequently, especially in collaborative annotation situations, users also need to search for results of co-annotators, experts, or other sources [HHD08]. The interview at the IPN Kiel (Section 4.1.2) revealed that novice annotators use already analyzed videos as training material and compare their own results with the results of their expert colleagues. To give another example, *CoScribe* enables annotators to compare own document structures with those of co-annotators [SBM08] (cf. Section 4.1.2).

Exploration of co-annotators' data also can be an issue in asynchronous collaborative projects which proceed over a long timeframe. After being absent, users may need to track the changes performed by other annotators (or annotaion tools) involved in the project. In this context, the term *Collaborative Awareness* was introduced in Section 3.3.7. This includes that individual users working within a group obtain information about the presence of co-working users. To take an example, *AnnoCryst* notifies users through RSS Feeds when other group members have made changes on a related data set. In addition to that, users may be enabled to browse chat or commentary histories [BFW07]. Obviously, tracking of changes can be a relevant tasks in synchronous collaboration settings.

Exploration also includes restructuring of the data representation. With regard to this, annotators are allowed to contrast relevant data with each other, or to hide less important information. For this purpose, activities of searching, filtering, sorting (including grouping, clustering, and hierarchization), or item relocation by means of direct object manipulation [Shn97] have to be performed (which is supported by a majority of the analyzed annotation systems). This is especially important when annotators are confronted with a large amount of annotations including those of co-annotators and other external resources [CCG02, HHD08]. In content analysis, pooling commonly classified information and making statistical comparisons are part of re-composition [PH07, SPK05]. According to this, exploration also supports reflection. Thus, it facilitates the consideration of multiple views of the video where users are allowed to obtain perspectives on the contents beyond their subjective point of view [MB04, SFZ06, VSN05].

5.2.7 Externalization

The *Externalization* phase refers to three different aspects at the end of the annotation process: Publication of the process results, creation of summaries, and export of data in order to conduct post-processing with external applications.

Publishing the results of an annotation project can have different goals. Usually, it begins with editing and converting the data into several formats, and moves on to presenting this information by means of corresponding media [PH07]. Published results can be used for demonstration purposes. Reporting on the CPV Video Study Physics (cf. Section 4.2.1), Mikova and Janik [MJ06] explain that exclusive video sequences of filmed lessons were later shown in teacher-training as examples for "good teaching". Also databases of already annotated material can serve as digital resource for information retrieval in following annotation sessions, e.g., comparable to the preceding training phases conducted by the IPN (Section 4.1.2), in which novices explore already analyzed video sequences [SPK05].

Moreover, a goal of annotation can be obtaining (mostly automatically) generated surveys and assemblies of similarly categorized content subsets. Creating surveys, assemblies, and summaries is an elaborate and time-consuming work. Thus, a lot of research is concerned with automatic approaches. For instance, Bertini et al. [BBN05] present a system that supports automatic extraction of video features as basis for semantic annotation in order to permit the automatic generation of summarizes of sport game highlights. The *eMargo* system provides functionalities to display only the contributions of one student in a single view. In doing so, teachers are facilitated by reviewing students data in order to conduct evaluation of accomplishment [SGRH07]. As a conclusion, externalization can be means of supporting quality control for annotation.

Furthermore, it is often necessary to export data for further processing by means of more specific applications. For example, experts at the IPN report on exporting data to various formats, such as tab-delimited text or transcription files, in order to generate statistical calculations with SPSS [SPK05]. Thus, further analytic activities can be executed with tools and services that are not provided by the applied annotation application. In addition to that, annotation tools such as *ATLAS.ti* or *EXMARaLDA* show that, due to the wide range of exportable file formats, interoperability between several familiar systems can be significantly fostered.

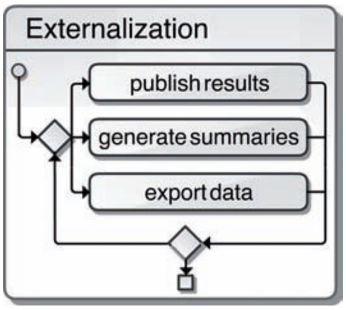


Figure 5.7: Externalization of Annotated Data.

5.3 Discussion: Validity and Applicability of the established Model

In this section, both the validity and applicability of the generic process model are discussed. Validity primarily refers to the generality of the model and is substantiated by the methodology applied for the preliminary analysis and studies. Furthermore, the applicability of this model will be disclosed by referencing the conceptual framework which has to be regarded as the technical solution model contributed by this thesis.

5.3.1 Validity of the Process Model

As stated above, the validity of the model is determined by the achieved level of generalization, that is, fundamental criterion is the factual consideration of diverse facets of multimedia annotation with regard to different concrete workflows, multiple functionalities and approaches, as well as the different forms of heterogeneous data which are generated and treated in the course of the annotation process.

According to these viewpoints, generality has been ensured through the approach selected for the preliminary investigations. Thus, the feature-oriented analysis of annotation systems included ten applications which represent different classes of multimedia annotation systems, previously defined according to given application domains and supported media formats (cf. Appendix A). In this context, functionalities as well as related data formats were summarized which arise from different areas of multimedia annotation. Moreover, the identified features were connected to classes of tasks and activities performed during annotation. These classes relate to a specific category system, whose universal validity is given by its inductive development in the course of the entire analysis.

The generic characteristics of incorporated phases and sub-processes and specified sequential relations was approached by the consideration of different concrete workflows of multimedia annotation. On a praxis level, this has been guaranteed by the questioning and observation of experts from five heterogeneous domains, and by summarizing located various field reports and proposals for project execution. On a theoretical level, different use case specific models of annotation workflows were taken into account.

5.3.2 Applicability to a Solution Concept

Aspect of applicability refer to the transfer of the obtained findings to a technical concept. Hence, reasoning can be performed based on fundamental elements of the *Conceptual Framework for Process-based Multimedia Annotation* which is elucidated in Chapter 6.

In Section 6.2, a formal specification of the annotation process is described, which constitutes a reference specification for the modeling of annotation workflows by means of an arbitrary process definition standard. This has to be regarded as a formalized description of the proposed process model, since it determines the structural organization of defined tasks, sequences, and services required in a specific use case.

In the scope of the definition of a reference architecture model (see Section 6.3), various architecture components are incorporated as enclosed functional units. As will be described in Section 6.3.4, components of one of two distinguished classes correspond to sub-processes of annotation described in this chapter. Here, especially the identified forms of data which are transfered between different tasks and services form the basis for an abstract definition of required interfaces.

Furthermore, this chapter clarified that annotation processes, especially in the scope of collaboration, can mostly not be regarded as pipeline-like procedures. It has been showed that returns and iterations can be required, which may result from explorative activities such as comparison of further results, and might purpose data improvement or adjustment. According to this, conclusions can be drawn with respect to the applied visualization model in order to represent annotation workflows, as well as for aspects of interaction regarding facilities to manually intervene in the running annotation process.

5.4 Summary

In this Chapter, a Generic Process Model of Multimedia Annotation has been presented. Here, detailed knowledge is provided about (i) workflow phases, sub-processes, and tasks, (ii) operational interrelations between these aspects, (iii) services (tools, functionalities, approaches, etc.) that are exploited in order to accomplish annotation tasks, and (iv) forms of data that result from service application or are transfered between different steps of the annotation process. A summary of the aspects comprised in the process model is provided in Figure 5.8.

In order to obtain this knowledge, an examination of annotation practices was conducted by performing an empirical study and a survey of related literature. The empirical study was held at five research institutes, at which annotation is applied in every day praxis. Here, researchers or teaching staff were interviewed and observed during the operation of the respective annotation system. The literature survey included existing models, reports on conducted annotation processes, and proposals of how to perform annotation. The findings were combined with results of a feature-oriented analysis of ten annotation systems. While the conducted analysis aimed at identifying tools and functionalities provided by common applications as well as forms of data that are generated, the empirical study and literature survey revealed the operational relations between these services and data with respect to execution sequences, transitions, jumps, and loops within the process. Here, a relevant conclusion was that it is not possible to define "best practices" of annotation, since the concrete workflows differ from each other, depending on the purposes, objectives, supported media formats, and application domains at which annotation is performed. Consequently, the presented process model has been specified on an abstract level and is thus generic, so that different instances of concrete annotation workflows can be mapped.

The established process model for multimedia annotation has been elucidated in Section 5.2. It includes three general phases: *Preliminary*, *Generative*, and *Subsequent*. The *Preliminary* phase refers to activities that are conducted before effectively starting annotation. It consists of *Scheduling* tasks, at which the project is planned and the required data is gathered, as well as *Configuration* tasks which concern all settings that have to be adjusted in order to prepare the employed annotation system. Among these settings are user interface configurations, management of users and incorporated tools, assignment of groups, roles, access rights, and tasks, as well as annotation-related properties such as the applied annotation method or scheme.

A *Generative Phase* of an annotation process regards all activities that are associated to, de facto, activities of annotation. Here, sub-processes of *Selection*, *Addition*, and *Exploration* have been localized. In particular, it has been pointed out that these sub-processes are strongly interrelated. That is, annotators select areas-to-annotate, add new information, and need to search and browse in mutual manner. *Selection* comprises the definition of areas-to-annotate within the media documents. Here, structural metadata is generated that indicates for an entire document, or spatial, temporal, or spatio-temporal areas within a document. Furthermore, *Addition* was identified as the sub-process which comprises tasks of information attachment. According to the different forms of annotations illustrated in Section 2.1.3, information attachment tasks have been classified according to different types of annotation that are going to be attached: metadata, (multimedia) content, dialog acts, or hyperspace elements. Metadata can be attached in order to classify contents, describe media content features (image colors, music chords, etc.), or to attach metainformation (author, title, date, etc.). Attachment of multimedia contents may have different objectives, too. These include transcription of communication or behavior, description of observed facts, interpretation and evaluation of such facts, or general contribution of comments and notes. In the scope of collaborative annotation processes, the addition of dialog acts is an instrument of

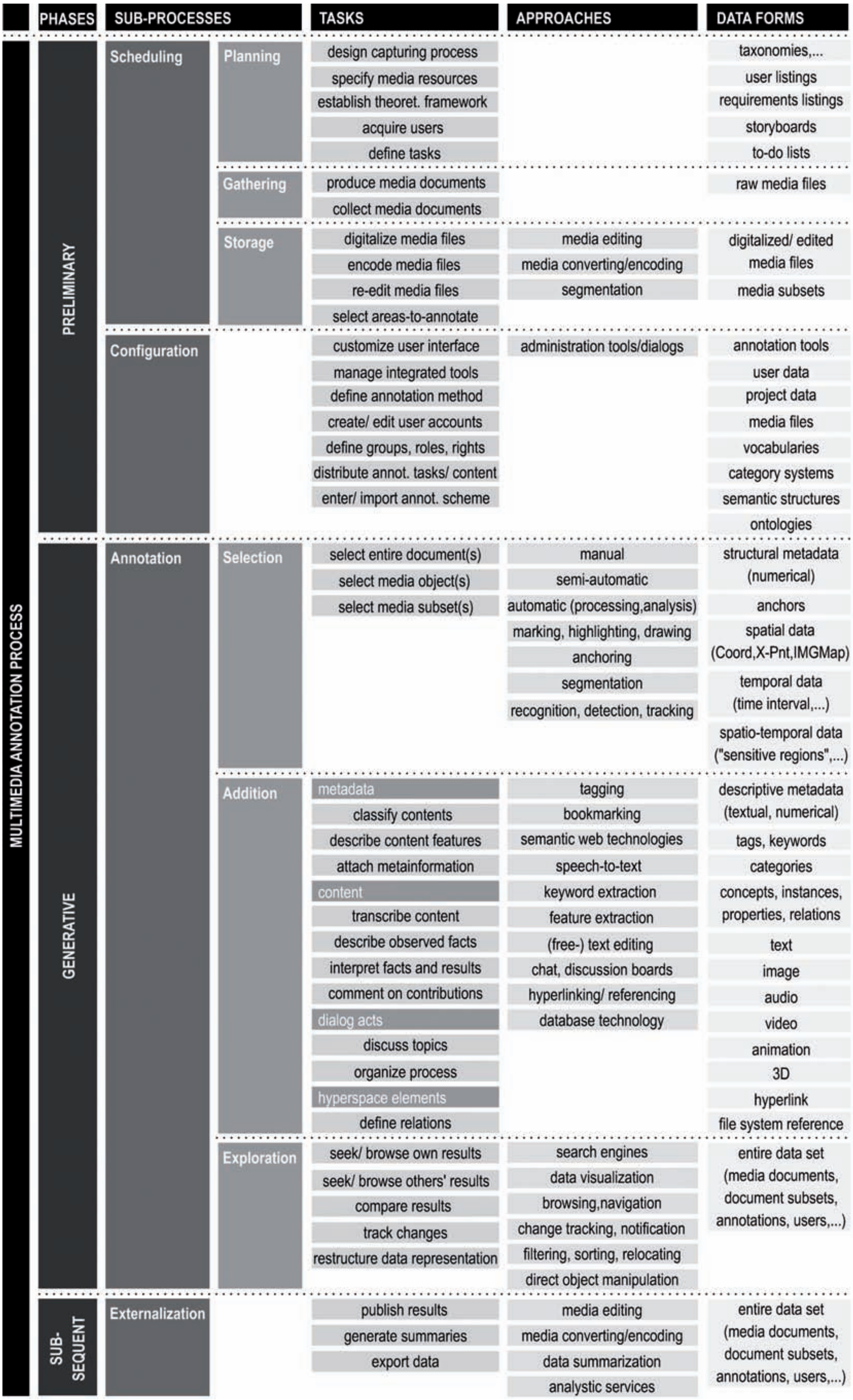


Figure 5.8: Multimedia Annotation Process: included Phases, Sub-Processes, Tasks, Approaches, and Data Forms

discussing topics around media objects and annotations, as well as coordinating the common annotation project by clarifying organizational issues. Finally, relations can be defined, for instance, by specifying hyperlinks, in order to build networks or simply to connect external (multimedia) resources with existing data. As stated above, *Exploration* activities escort activities of selection and information addition during the entire generative workflow phase. Here, annotators seek, browse and navigate through own and external data, such as results of co-annotators. In this context, the comparison of annotation results, but also the repeated review of generated data leads to returns to already accomplished phases or tasks of the project. Moreover, it has been identified that, when working in teams, users need to track changes made by other participating persons (or tools). Exploration also includes activities of representation manipulation. That is, users filter, sort, or relocate data items, also by means of direct object manipulation.

The last phase, here called *Subsequent*, relates to all tasks and activities that are to be performed after effectively annotation has been accomplished. In this project stadium, *Externalization* tasks are conducted. Among these are (i) the publication of results from annotation, which is often connected with re-editing and encoding activities, (ii) the generation of summaries in accordance with specific criteria, or (iii) data export, e.g., in order to reuse annotation data in other projects or with further applications.

The presented Generic Process Model of Multimedia Annotation and its including results form a groundwork for a process-based design of multimedia annotation systems. Accordingly, these findings were used for the development of a *Conceptual Framework for Process-based Multimedia Annotation*, which will be elucidated in the following chapter.

Chapter 6

A Conceptual Framework for Process-based Multimedia Annotation

This chapter presents a solution concept for the design of process-based multimedia annotation systems. This concept is particularly based on the general goals of this thesis, which have been explained in Section 1.2 referring to an improvement of the interactive operation of annotation systems.

In this context, the first goal refers to a provision of specific process-related information about (i) the entire set of annotation tasks that have to be accomplished, (ii) the workflow procedure regarding the sequential arrangement of annotation tasks, (iii) the current task to perform, (iv) the given functionalities, tools, components, or services which can be applied in order to process the current task, and (v) the actual workflow state with respect to already accomplished tasks as well as steps which lay ahead.

In addition to these aspects, especially with regard to point (iv), a second goal is to explicitly supply the appropriate services (i.e., respective UI representatives) in the right scope or moment on the graphical user interface, according to the recent annotation task to be processed. In contrast, services which have to be regarded as not relevant in the scope of a certain workflow state are to be hidden or closed.

Third, in consideration of a multitude of different potential annotation workflows (cf. Section 4.3), facilities are to be provided for an individual case-specific definition of annotation workflows by means of standard process definition approaches.

According to the structure presented in Figure 6.1, the developed conceptual framework can be subdivided into three constituent parts, covering different aspects and layers of a respective technical solution: a *Data Modeling and Management Concept*, an *Architectural Concept*, and a *Visual-interactive Concept*.

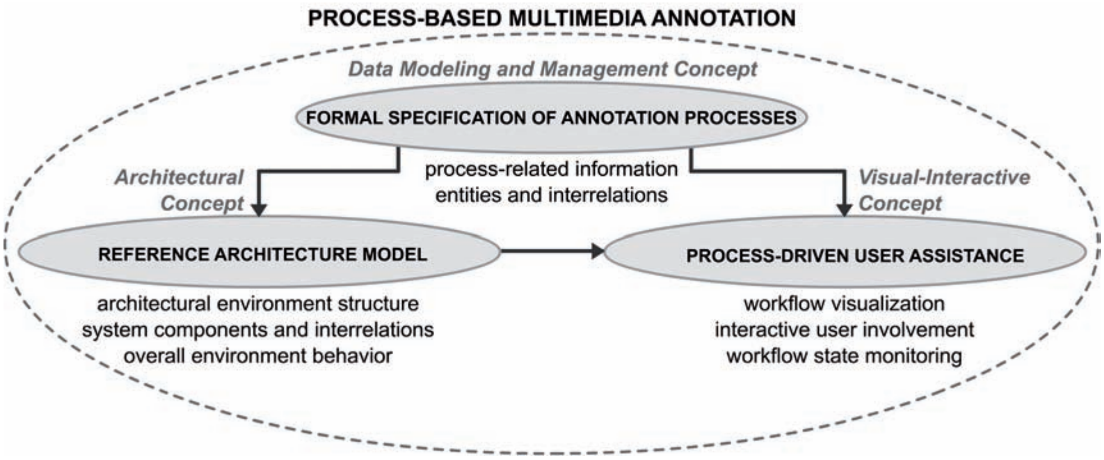


Figure 6.1: Constituent parts of the established Conceptual Framework for Process-based Multimedia Annotation.

Data Modeling and Management Concept. Initially, a *Formal Specification of Multimedia Annotation Processes* is constituted. Here, the fundamental elements comprised in an annotation process are elucidated, defining basic classes of information entities (and respective interrelations) that are to be processed and managed by a process-based annotation system. Additionally, guidelines are provided with respect to the definition of annotation processes by means of workflow modeling standards. Hence, a data modeling concept is developed that serves as initial groundwork for the consecutive partial concepts, determining the process-related information processed within a respective system architecture, and also visualized and interactively modified by human workflow participators upon a graphical user interface.

Architectural Concept. In the scope of an architectural concept, a *Reference Architecture Model* is established. This model includes the basic structure and organization of all components included in the entire environment. Furthermore, incorporated basic components are described, defining respective functions, interfaces, and the interaction and data exchange between different components on an abstract level. In doing so, the overall behavior of the framework is determined. Consequently, a structural technical basis is constructed, on which developed approaches associated with the system user interface layer can be superimposed.

Visual-interactive Concept. Regarded as central contribution of this thesis, a visual-interactive concept is presented which aims at realizing *Process-driven User Assistance*. In general, this partial concept includes the visualization of workflow-specific information which has to be declared as relevant for users at workflow execution time. In this context, facilities for an interactive involvement of users in a running workflow performance are constituted. Moreover, an approach is presented in the scope of visual workflow state updating for monitoring purposes, considering specific challenges of multimedia annotation and collaboration.

The remainder of this chapter is organized as follows. In Section 6.1, a set of requirements will be derived with a special focus on the specific characteristics of annotation workflows and collaborative usage scenarios. Then, according to the distinction of different partial concepts, a *Formal Specification of Multimedia Annotation Process* will be constituted in Section 6.2. Section 6.3 addresses characteristics and components of a developed *Reference Architecture Model*. Subsequently, a concept for *Process-driven User Assistance* will be elucidated in Section 6.4. A summary of the main results of this chapter is given in Section 6.5, including a verification for the fulfillment of specified requirements.

6.1 Derivation of Process-related Requirements

Conceptual design decisions are particularly based on the endeavor to support given operating procedures within the annotation process. In this context, basic requirements can be identified that have to be fulfilled by a process-based annotation framework. Up to now, several research papers have presented requirements for the design and development of annotation systems [GVOH05, HB92, HJK$^+$09]. Nevertheless, presented aspects particularly refer to more general topics, such as tools and functionalities that need to be provided, or usability-related aspects. As an enhancement of existing work, specific requirements are defined in this section which result from an explicit consideration of annotation workflows. In addition to that, further aspects are elucidated that can be derived from collaborative usage scenarios.

The derivation of specific requirements is based on the general goals pursued in the scope of this thesis. As described in Section 1.2, these objectives can be summarized as follows:

1. Automatic execution of the annotation workflow.

2. Provision of visual and interactive access to a given annotation workflow, including the involved tasks, their sequences, as well as the recent workflow state.

3. Explicit supply of the suitable annotation tools or services, depending on the recent task to perform.

4. Individual specification of concrete annotation workflows in order to support multiple use cases.

In addition to these goals, the consideration of collaborative work settings lead to further relevant requirements. The assignment of derived requirements to specified goals is summarized in Table 6.1. The determined process-related requirements are described in the following paragraphs.

Goals / Viewpoints	Requirements
Workflow Execution	R2: Workflow Control R3: Sub-Process Enclosure R6: Data Consistency. R7: Coherent Data Junction and Management
Workflow Visualization	R9: Process Visualization and Monitoring R10: Interactive Involvement of Users
Explicit Service Supply	R2: Workflow Control R3: Sub-Process Enclosure R4: Flexible Service Integration R5: Workflow-driven Service Invocation
Workflow Specification	R1: Process Definition R3: Sub-Process Enclosure
Collaborative Work	R6: Data Consistency R8: Workload Distribution R11: User Management R12: Data Exchange and Sharing R13: Correct Data Handling

Table 6.1: Derivation of Process-related Requirements based on Goals and Viewpoints pursued in this thesis.

6.1.1 Process-related Requirements

R1: Process Definition. The conducted studies on annotation practices presented in Chapter 4 revealed that a multitude of different annotation workflows are applied in praxis, depending on the goals and application domains in which annotation is employed. Consequently, in order to support different concrete workflow instances of annotation, the conceptual framework must provide facilities to predefine individual workflows by means of a standardized process definition languages (cf. Section 2.2.3). Here, different process definition languages need to be supported. Within a process definition, several properties are defined, such as the tasks that have to be worked on, task execution sequences, conditions that are tied to such task transitions, and the workflow participants, i.e., users and tools associated with the tasks. In order to provide guidelines for the definition of annotation process instances, a formal specification has to be developed in the form of a *workflow scheme*. This scheme serves as template that determines the incorporated elements and the basic structure of the process definition.

R2: Workflow Control. Workflows of collaborative multimedia annotation can be complex, since they might be network-like and potentially contain several iterations. Thus, transitions between workflow phases and the control of sequences of sub operations have to be supported. Also loops and re-entries to other phases of the workflow must be considered. Hence, the framework needs to include a specific control unit which realizes all procedures with respect to workflow control. For instance, this unit is responsible for execution handling, that is, it has to decide which is the next task and accordingly invoke the correct system components. For this purpose, a control unit needs to interpret previously specified process definitions. With respect to the proposed framework, workflow control forms the basis for appropriate *process visualization* and *user assistance*. In this context, a seamless usage of an annotation system can be guaranteed.

R3: Sub-Process Enclosure. In order to foster workflow control procedures, subprocesses of annotation must be made "tangible" for a respective control instance. Thus, identified items included within the generic process model of annotation illustrated in Section 5.2 need to be pooled into functional units that are mutually delimited. In doing so, task areas can be typecasted and invoked by addressing respective modules. According to the assignment of annotation sub-processes to enclosed entities, it is also crucial to provide specific interfaces which govern the access to process-based modules, as well as the forms of associated input and output data.

R4: Flexible Service Integration. Typically, users perform activities that may require different forms of annotation. This requires additional flexibility to an annotation system, which should seamlessly support the different ways of using annotations. As a consequence, the system must avoid tying the annotation to a fixed set of predefined annotation types [ABDF07]. In this context, Agosti et al. [ABDF07] and Constantopoulos et al. [CDTT04] note that flexibility in order to support different ways of annotation as well as integration of different services are key aspects of the design of annotation systems. Thus, system flexibility is required that allows the integration of arbitrary annotation services in order to support different forms of annotation (regarded as data and as process). Such flexibility requires a framework that provides interfaces or "docking ports" for multiple tools or external services. In addition to that, the system must enable administrators to integrate, replace and remove tools that can be assigned to task-related modules. That is, a specific management of integrated services has to be realized.

R5: Workflow-driven Service Invocation. As a consequence of the integration of an undetermined number of annotation services, a large number of potential available functionalities can lead to complex and multi-optional user interfaces that obstruct especially unexperienced users. According to one of the central principles of *Workflow Management* (see Section 2.2.1), usability can be improved by proposing or invoking the required service(s) at the "right time" or in the "right scope", depending on the task which has to be processed at the current workflow state. Hence, workflow-driven service invocation has to be regarded as a key requirement with regards to a concept for *Process-driven User Assistance*, which will be proposed as visual-interactive contribution of this thesis in Section 6.4.

R6: Data Consistency. Consistency of the handled set of data is required in two ways. First, with respect to the flexible integration of different annotation tools, it is assumed that multiple components permanently read and possibly write on the same data. Thus, the consistency of shared parts of the data set has to be ensured at every point of the annotation process. Second, with respect to collaborative practices in which data is exchanged or shared between members of a community, consistency of data must also be warranted for every peer that participates in the shared system.

R7: Coherent Data Junction and Management. Besides of consistency of data, the integration of different components requires specific strategies for data junction and management. This can be attributed to the following factors. Integrated tools are responsible for the treatment of different kinds of annotation tasks, such as segmentation, metadata generation, or commenting. Particularly, as clarified in Section 5.4, different forms of data are handled with respect to media formats of the annotated documents, as well as the generated annotation data, such as structural or descriptive metadata, multimedia content, or references to external files or resources. Here, standardized formats for multiple kinds of annotation data are still lacking [HJK^{+}09]. Thus, resulting requirements refer to the storage and management of a heterogeneous data set, at which information coded in differing formats need to be centrally pooled. For this purpose, the proposed framework has to provide a generic data model in which processed information is integrated in appropriate manner. In doing so, relevant aspects of annotation processes are supported, such as export or publishing in standardized formats (cf. Section 5.2.7), or a synchronized representation and visualization of information at run-time (cf. Section 2.1.4).

R8: Workload Distribution. Volkmer et al. [VSN05] constitute relevant requirements in order to support annotation workflows performed in collaborative settings. Thus, in order to make the annotation process seamless and user-friendly, annotation must be structured to tasks that are then assigned to the different participating annotators. Moreover, they point out the requirement of dynamically adjusting these sets of work to accounts to counteract unexpected relocation of resources. As a result, an account-based task and range assignment, i.e., the distribution of the given *workload*, must be realized. Thus, collaborators can be provided with an improved overview of the tasks they have to perform in the current phase of the project. Furthermore, a regulation of the collaborative annotation process can be realized, e.g., by enabling a group administrator to distribute workloads to be attended by other members (cf. [yLTS03]). Basically, the term workload can be regarded from two different views. First, it might refer to the number of different tasks that have to accomplished, such as segmentation, metadata generation, or commenting. Second, also the set of media documents to annotate can be distributed to individual users or groups.

R9: Process Visualization and Monitoring. With increasing complexity of the graphical user interface of an annotation environment concerning available features and the number of tasks that potentially need to be accomplished, users progressively struggle with system operation, particularly with respect to the following questions: "What is the current state?", "Which is the recent task?", "Which are the consecutive steps?", and "What has already been completed?" According to this, a visualization of the recent annotation process has to be provided which represents a previously generated process definition in graphical manner. Particularly, the specified tasks as well as their executive order need to be in focus. In addition to that, options have to be offered which enable monitoring of the annotation process. This especially applies to the distribution of the annotation workload. Moreover, with regard to workload distribution issues in connection with account-based user management, a personalized view on the annotation process must be provided for varying roles of users and groups (see also [BRB07, BRB05]).

R10: Interactive Involvement of Users. The conducted *Studies on Annotation Practices* described in Chapter 4, as well as a *Generic Process Model of Multimedia Annotation* established in Chapter 5 have revealed that a (collaborative) annotation process might be characterized by several returns to previous workflow steps, loops, and iterations. Frequently, the reason for this is that generated data potentially needs to be re-edited, for instance, as a result of comparative exploration activities. In addition to that, previous studies on user guidance in comparable application areas revealed that a strong guidance (if well-implemented) can be helpful for novice users, but more experienced persons rather prefer the admission of more degrees of freedom with respect to task selection and execution [SBCO01, ZZZ07]. Thus, strong user guidance can lead to negative effects with respect to performance factors as well as user satisfaction. As a result, a solution concept must consider methods for an interactive involvement of human workflow participators referring to the workflow procedure, provided on different levels of user guidance.

R11: User Management. Especially in the scope of collaborative work settings, specific user management is required. That is, users have to be assigned to accounts, user groups, and roles with respective access rights. Thus, User Management has to be regarded as indispensable for workload distribution purposes. Moreover, it forms the basis for a personalized visualization of the annotation process, considering different potential user groups and roles.

R12: Data Exchange and Sharing. In collaborative use cases, the annotation environment has to realize data exchange and sharing between multiple users that are potentially separated over space. For this purpose, stored information must be made available to every participant of the group. Usually, this is implemented by the employment of client-server environments, peer-to-peer networks, specific broadcasting mechanisms, or simply email. With respect to collaborative annotation systems, related data includes particularly the processed media files (or references to them) and all generated annotations. In addition to that, with a focus on the specific aspects of this thesis, stored data also needs to include user-defined process definition files as well as listings of available integrated services (e.g., in forms of URLs or API specifications).

R13: Correct Data Handling. In the scope of collaborative work, in which each group member has full access to shared contents, Stahl et al. regard the emerged information set as a *Dynamic Information Space*. That is, the information set is continually changed and extended [SFZ06]. In addition to that, annotations can be assigned to different kinds of contents. Thus, an appropriate handling of media files, its annotated information, as well as their organizational structure must be provided.

6.1.2 Summary

In this section, a set of requirements has been derived which especially focuses on the design of process-based multimedia annotation systems, also considering the application in collaborative work settings. In the following, according to the first listed requirement *Process Definition*, basic information entities related to processes of multimedia annotation will be presented by means of a formal specification. A verification for the fulfillment of specified requirements by the conceptual contributions of this thesis will be conducted in Section 6.5.

6.2 Formal Specification of Multimedia Annotation Processes

As already stated in Section 2.2.3, a workflow can be regarded as a business process that is executed by means of computing devices [Sch00]. Such processes are automatized, and documents, tasks, or contents are transfered according to predefined rules [WFM99]. Basically, a process-based system must be driven by a specific process model [LAH08]. According to this, the concrete applied processes must be modeled and fed into the framework in forms of a *workflow scheme* or *workflow specification* [GKK+08]. Here, workflow schemes describe a process by means of a workflow modeling language for the purpose of documentation, analysis, or execution within a workflow management system [Sch00].

In this section, a formal specification of multimedia annotation processes is constituted, including basic classes of information entities contained in an annotation workflow. In this manner, a (multimedia) annotation-specific workflow scheme is established, providing guidelines for process modeling activities on the one hand, and defining all relevant process-related information that have to be handled by a process-based annotation system on the other hand. In the following, the basic process-related information entities and their interrelations will be elucidated, including the following elements: (i) *annotation process*, (ii) *work item*, (iii) *transition*, (iv) *media content*, (v) *annotation*, (vi) service, (vii) *user*, and (viii) *user group*.

6.2.1 Specification of Process Elements and Interrelations

On the formal specification of annotation processes, a set of different classes of process elements are distinguished by drafting definitions which describe the basic properties, sub-elements, and relations to other process entities. Initially, a definition of the entire annotation process is given. In the course of this section, further definitions are constituted which provide more detailed explanation of the incorporated process instances.

Definition 1. *An annotation process is defined by a tuple* $p = (i_p, W, T, M, A, C, S, U)$ *where:*

- i_p *is the unique identifier of* p.
- $W = \{w_1, w_2, \ldots, w_n\}$ *is a finite set of work items which assemble* p.
- $T = \{t_1, t_2, \ldots, t_n\}$ *is a finite set of transitions located between consecutive work items.*
- $M = \{m_1, m_2, \ldots, m_n\}$ *is a finite set of media contents which are processed in the course of the execution of* p.
- $A = \{a_1, a_2, \ldots, a_n\}$ *is a finite set of annotations which are generated, edited, and transfered between work items in the course of the execution of* p.
- $C = \{c_1, c_2, \ldots, c_n\}$ *is a finite set of conditions which are connected to the entry or termination points of work items.*
- $S = \{s_1, s_2, \ldots, s_n\}$ *is a finite set of services which are offered to perform the single work items included in* p.
- $U = \{u_1, u_2, \ldots, u_n\}$ *is a finite set of users participating in* p.

This definition aggregates the basic elements of an annotation process. Among the basic process elements are all work items, which represent the annotation tasks to perform. Moreover, transitions between different work items are managed, determining the general overall procedure of the process. Further entities are the multimedia contents as well as respective annotations generated and modified during the workflow execution. Besides of transitions, also specific conditions define the sequential execution of the process. Here, conditions are usually defined by an indication of code parts, interfaces, or APIs. The executable work items are connected to specific annotation services, at which the entirety of all available services is managed on process level, since multiple assignment must be facilitated. Finally, all participating users are managed and provided with specific properties (discussed later on, see Definition 7).

Definition 2. *A work item is defined by a tuple* $w = (i_w, i_{pw}, wt, ws, S_w, M_i, M_o, A_i, A_o, C_i, C_o) \in W$ *where:*

- i_w *is the unique identifier of* w*.*
- i_{pw} *is the unique identifier of the parent work item of* w *(optional).*
- $wt \in WT : W \rightarrow WorkItemTypes := \{Configuration, Selection, Addition, Exploration, Externalization\}$ *assigns a work item type to* w*, indicating for the type of annotation task performed by a human or technical participator.*
- $ws \in WS : W \rightarrow WorkItemStates := \{NotEnabled, Enabled, Active, Accomplished\}$ *indicates for the current state of work item* w*.*
- $S_w \subseteq S = \{s_{w1}, s_{w2}, \ldots, s_{wn}\}$ *is a finite set of one or more services which are offered to perform* w*.*
- $M_i \subseteq M$ *is a finite set of media contents which are transfered to* w *as input data (optional).*
- $M_o \subseteq M$ *is a finite set of media contents which are forwarded by* w *as output data (optional).*
- $A_i \subseteq A$ *is a finite set of annotations which are transfered to* w *as input data (optional).*
- $A_o \subseteq A$ *is a finite set of annotation which are generated by* w *and forwarded as output data (optional).*
- $C_i \subseteq C$ *is a finite set of input conditions which are checked at the entry into* w *(optional).*
- $C_o \subseteq C$ *is a finite set of output conditions which are checked at the termination of* w *(optional).*

A work item is the representative of the work that has to be performed by a workflow participant [WFM99]. Thus, it has to be regarded as an annotation task that might be comprised of several subordinated activities or interactions. Here, an interleaving of work items is possible, which is realized by indicating for a parent work item. In doing so, *activity blocks* can be defined, which are defined as sets of work items that share common properties [WFM99]. Furthermore, different work item types can be specified. On an abstract level, work item types can be classified on the basis of the specification of annotation sub-processes conducted in Chapter 5, distinguishing between *Configuration*, *Selection*, *Addition*, *Exploration*, and *Externalization*. As will be described in Section 6.4.2, the state of a running annotation workflow undergoes permanent changes during its execution. In this scope, also the state of work items is modified. This is particularly relevant in the case of work items that allow a manual invocation by human participators. In order to realize the invocation of annotation functionalities in the right scope or moment with respect to the recent annotation task (such as will be described in Section 6.4.3), each work item is connected to one or more respective services. In addition to that, work items (or connected services) obtain specific input data and pass generated output data back to the entire framework after its individual execution. Basically, this data includes media contents and annotation objects. Moreover, preliminary conditions determine whether the execution of a work item is legal depending on a specific workflow state. Output conditions can be defined which determine whether a work item can be transfered into an *accomplished* state.

Definition 3. *A transition is defined by a tuple* $t = (i_t, i_o, i_d) \in T$ *where:*

- i_t *is the unique identifier of* t.
- i_o *is the unique identifier of the origin work item, from which the transition* t *is launched.*
- i_d *is the unique identifier of the destination work item, at which the transition* t *targets.*

Basically, transitions connect different work items and thus define the overall workflow procedure on a general level (without the consideration of conditions checked at workflow run-time). Usually, process definition standards include unidirectional transitions. Hence, a transition is characterized by one origin work item and one destination work item. Since transitions are managed on the entire process level, work items can be assigned to multiple incoming and outgoing transitions.

Definition 4. *A media content is defined by a tuple* $m = (i_m, i_{pm}, l_m, mt) \in M$ *where:*

- i_m *is the unique identifier of* m.

- i_{pm} *is the unique identifier of the parent media content of* m *(optional).*
- l_m *indicates the location of* m.
- $mt \in MT : M \rightarrow MediaContentTypes := \{Text, Image, Audio, Video, 3D, OtherFormat, ComposedDocument\}$ *assigns a media content type to* m.

Media content entities implement the different types of media documents processed in the course of the annotation workflow. First of all, several media formats can be distinguished, such as text, image, audio, video, 3D, or other formats. Additionally, where necessary, it has to be indicated whether a given media content is a composed document. Examples for composed documents are websites or PDFs including subordinated media objects such as text, images, etc. In this case, the specification of a parent media content is required within included sub-contents.

Definition 5. *An annotation is defined by a tuple* $a = (i_a, i_{pa}, at) \in A$ *where:*

- i_a *is the unique identifier of* a.
- i_{pa} *is the unique identifier of the parent annotation of* a *(optional).*
- $at \in AT : A \rightarrow AnnotationTypes := \{ValidityArea, Metadata, MediaContent, DialogAct, HyperspaceElement\}$ *associates* a *with an annotation type.*

In Section 5.4, it has been clarified that several forms of annotations might be generated, edited, and managed in the course of the annotation process. In general, among these are *validity areas*, which include structural metadata, defining anchors or the connection points of other annotations. Moreover, annotations can be descriptive or semantic *metadata*, *media contents* or subsets, *dialog acts* (mostly textual contributions), and *hyperspace elements* such as relations. In some cases, for instance, in the context of dialog acts, a hierarchical structuring is required, which is realized by the indication of parent annotations (see also Section 2.1.1 *From Paper-based to Digital Annotations*).

Definition 6. *A service is defined by a tuple* $s = (i_s, l_s, sd) \in S$ *where:*

- i_s *is the unique identifier of* s.
- l_s *indicates the location of* s.
- $sd \in SD : S \rightarrow ServiceDescriptions := \{WT_s, MT_s, AT_s\}$ *assigns a finite set of service descriptions to each service* $s \in S$*, indicating for what kind of work item types* $WT_s \subseteq WT$ *and what kind of media content types* $MT_s \subseteq MT$ *service* s *can be applied to, and what kind of annotation types* $AT_s \subseteq AT$ *are generated.*

Services are to be regarded as the basic system features that are invoked during the annotation workflow in order to perform specific tasks. In general, a service can represent a tool, a software component, an addressed stand-alone application, or an external service such as a web service. The latter case shows that it is indispensable to indicate for the exact location of a service. Furthermore, in order to ensure a correct service integration with respect work item assignment and the association with the adequate parts of a data model applied by the annotation system, a proper service description is required. This especially applies to extensible systems concerning the flexible integration of features. Here, the fundamental informations are the types of annotation tasks (work items), multimedia contents, and annotations supported by the service. In doing so, information about the provided functions, produced data, as well as expected data can be accessed by the framework.

Definition 7. *A user is defined by a tuple* $u = (i_u, G_u, R_u, W_u, M_u) \in U$ *where:*

- i_u *is the unique identifier of* u.
- $G_u \in G$ *is a finite set of groups a user belongs to (optional).*
- $R_u \in R = \{r_1, r_2, \ldots, r_n\}$ *is a finite set of roles that are assigned to* u *(optional).*
- $W_u \subseteq W$ *is a finite set of work items which are to be accomplished by the user* u *(optional).*
- $M_u \subseteq M$ *is a finite set of media contents which are to be annotated by the user* u *(optional).*

As already stated in Section 6.1, especially collaborative annotation workflows require an explicit management of user-related information. Here, in the context of task and/or content distribution, user accounts might be assigned to predefined groups. In this scope, specific roles can be defined and associated, which particularly determine access rights and additional task assignment. In addition to that, task and content assignment can be regarded and managed as user-related *workload*. Accordingly, the user workload is defined by one ore more work items W_u and/or one or more media contents M_u that have to be processed by the user within the process. Here, the concrete determination of the user workload depends on specific factors related to the workflow run-time state. This issue will be elucidated in Section 6.4.2.

Definition 8. *A user group is defined by a tuple* $g = (i_g, U_g, W_g)$ *where:*

- i_g *is the unique identifier of* g.
- U_g *is a finite set of users belonging to* g.

- $W_g \subseteq W$ *is a finite set of work items which are to be accomplished by the group* g *(optional).*
- $M_g \subseteq M$ *is a finite set of media contents which are to be annotated by the group* g *(optional).*

A group comprises multiple human workflow participators which might share several common properties, for instance, a commonly assigned workload. Consequently, as with for individual users, an overall group workload can be specified, which is valid for all included group members.

6.2.2 Summary

In this Section, essential classes of annotation process elements have been specified. In doing so, a workflow scheme has been established, which can be regarded as guideline framework for the modeling of annotation workflows. This scheme can be transfered, for instance, into an XML-based structure, which is the basis for standard process definition languages such as BPEL or XPDL (cf. Section 2.2.3). In addition to these points, the fundamental information entities have been described which need to be handled by a process-based multimedia annotation system.

The specified basic process elements include (i) work items that have to be processed by human and/or technical participators, (ii) multimedia contents and annotations handled by the system during the process, (iii) specific conditions which determine the workflow procedure during run-time, (iv) the services (technological workflow participants) assigned to work items in order to accomplish certain annotation tasks, and finally (v) the human workflow participants which perform the entire annotation process or parts of it.

A graphical processing and interactive access to the presented process-related information entities is addressed in Section 6.4. In the following, an architectural concept will be defined, describing the handling of process-related data on the part of the technical concept of a system. For this purpose, the relevant technological framework elements and their interactions and behavior will be illustrated.

6.3 Reference Architecture Model

In this section, a *Reference Architecture Model* is presented, at which the organizational and logical structure of a process-based annotation environment is defined. Here, process-related requirements defined in Section 6.1, as well as the identified annotation sub-processes and their sequential relations (see Chapter 5) are considered. Basically, a reference model can be regarded as template for the development of solutions which refer to specific tasks or problems [Del05]. It contains the formal or semi-formal description of business processes, data structures, organizational structures, or editing rules [Sch92]. In this context, a system architecture describes the single elements contained by a technical system with respect to their type, functional properties, and their interaction [Sch92].

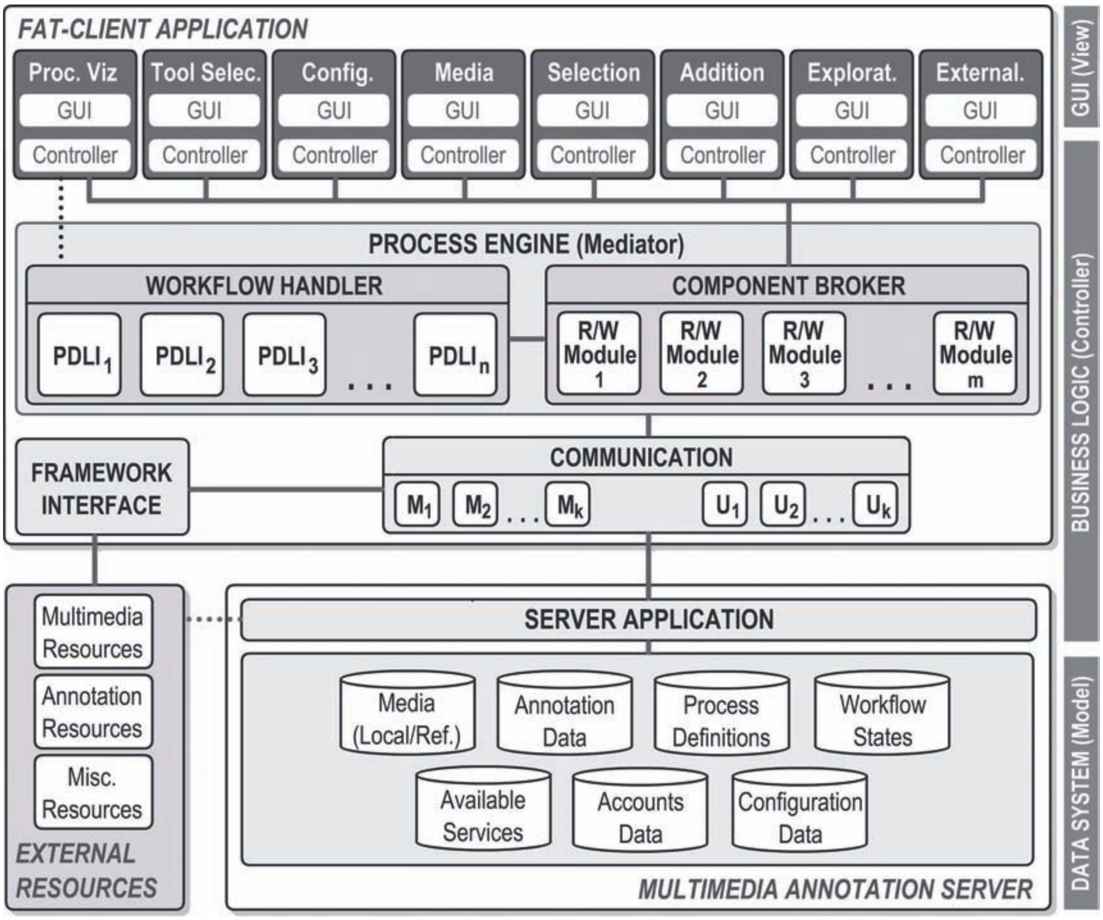


Figure 6.2: Proposed Reference Architecture Model for Process-based Multimedia Annotation Systems.

Figure 6.2 illustrates the proposed model, whose characteristics are elucidated in the following sections. These include a *Client-Server Model* (Fat-Client Application, Multimedia Annotation Server), which is described in Section 6.3.1. Section 6.3.2 explains the organization by an adapted combination of the design patterns *Model-View-Controller* (right figure margin) and *Mediator* (Process Engine). Moreover, the modularization through a *Component-based Environment* is elucidated in Section 6.3.3. In Section 6.3.4, the set of introduced components (top figure area) is described with a focus on their functions, procedures, and interactions. Section 6.3.5 provides an architectural concept in order to meet the specific challenge of data junction. Finally, the specified architectural subconcepts are summarized in Section 6.3.6.

6.3.1 Client-Server Model

A fundamental prerequisite for collaborative processes is the interconnectedness of every peer taking part for information exchanging purposes [BR04, Fin05]. In this context, a range of optional models that can be considered, such as *client-server*, *peer-to-peer*, or *web-based* approaches. Here, a client-server architecture is proposed due to its wide spreading in the area of information systems as well as specific services provided by server applications which are described in the next paragraph [BR04]. In addition to that, flexibility with respect to the replaceability of client as well as server applications can be guaranteed. For instance, client applications are enabled to communicate and exchange information with foreign servers providing annotated multimedia content.

A *Multimedia Annotation Server* realizes a centralization of the common information space. In doing so, the data system is available for any client that is connected. Additionally, it provides several services such as authentication of annotators, and management of accounts and access rights. In that context, modifications of the data set by the community are managed and provided to respective group members. The server is also responsible for a consistent storage and management of all relevant process data of the data system, which implements the *Model* of the proposed *Model-View-Controller Model* (MVC) explained below.

Basically, the client application handles user entries and interactions on the graphical user interface. It provides authoring options and features for media and annotation generation and editing. Here, a fundamental design choice is the realization of a so-called *Fat-Client* (or Rich Client). As opposed to *Thin-Clients*, rich client applications provide essential functionalities in order to realize specific computations without the use of server requests. Indeed, the boundaries between thin and fat clients are not explicitly defined [Kan98]. The client application is proposed to be implemented as Fat-Client, since different kind of functions can be assigned to the individual operation of the annotation framework as follows.

First, all relevant graphical components are offered which are to be regarded as part of the *View* layer of the MVC model. Here, either own graphical components are provided, but also external components or entire applications may be invoked and/or integrated. As will be explained below, each graphical component is connected to a specific controller which realizes local data control and management. In addition to that, process management is realized. That is, the interpretation, visualization, and execution of a loaded predefined process definition file is performed on client-side. In doing so, a concept for *Process-driven User Assistance* can be realized, which will be illustrated in Section 6.4. Finally, a specific interface is responsible for connection and communication with external resources, such as multimedia content providers or annotation services.

6.3.2 Adapted Model-View-Controller and Mediator

In order to realize an appropriate management of media files and its annotated information, existing approaches with regard to content annotation or linking have been considered in the scope of conceptual design processes. In the area of hypermedia research, several models can be identified, e.g. the *Dexter Hypertext Reference Model*. This model divides a system into three delimited layers, separating the data, the given annotation (hyper-) structure, and its representation on the graphical user interface [HS94]. The realization of a layer-based separation requires the definition of a specific architecture, which can be derived from a range of so-called *architectural patterns*. Such patterns are regarded as predefined descriptions of the composition, organization, and structure of a software system, and belong to the field of *design patterns* [BMR+96].

In the context of a conceptual framework for process-based multimedia annotation, an incorporation of the *Model-View-Controller Design Pattern* (MVC) [BMR+96, GHJV95] is proposed. As shown in Figure 6.3, it equally divides the application into three levels: the *model* layer represents the involved data, *views* display the information and assume user interaction, and the *controller* layer processes user entries and is enabled to modify data in the model. Furthermore, data consistency is warranted through a specific notification policy. In general, an MVC-based model can contribute to the following problems and requirements, which also fit to relevant aspects emphasized in this work (cf. Section 6.1): (i) implementation of interactive systems in general, (ii) separation of application data and functionalities, (iii) representation of the same information by different components, and (iv) immediate reflection of data modifications.

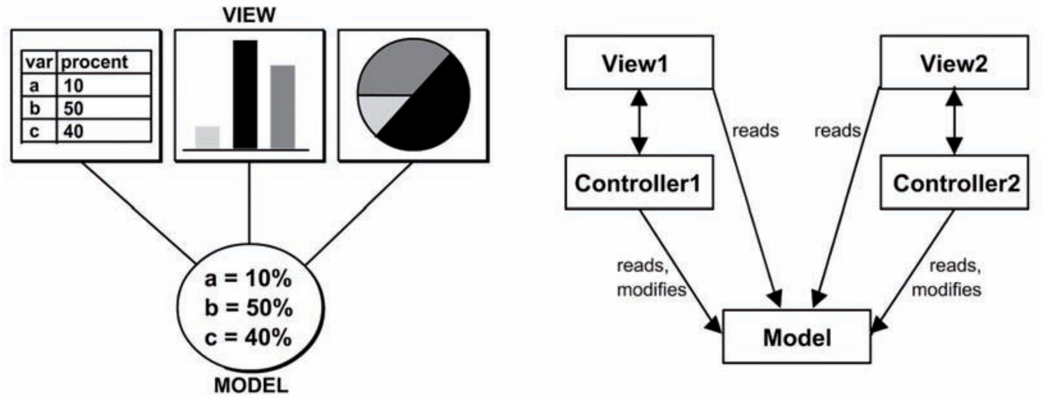


Figure 6.3: The Model-View-Controller Pattern. [BMR+96, GHJV95]

In the proposed reference architecture, the *Model* layer consists of data and information from contents and attached annotations to project configuration information, etc. In this context, the storage of user-generated process definition files and (references to) the available services constitute a central part. As such, this layer constitutes the *Data*

System of the proposed framework. As showed in Figure 6.2, the following types of data are included in the data system:

- *Media Files (Local or References).* In order to support collaborative use cases, sharing, exchange, and publishing is fostered by providing the multimedia contents on a common centralized resource (here, the Multimedia Annotation Server). Media files can be directly stored and made accessible, for instance by means of a file server. If external content is to be edited, such as web documents or contents provided by specific multimedia databases, references to these multimedia contents need to be managed. Referencing is usually realized by the specification of URIs, URLs, or pointers to the local file system.
- *Annotation Data.* The server is responsible for the centralized storage and management of all additional information generated during the annotation process. Among these data are structural metadata from selective sub-processes (like segmentation), descriptive metadata, media content (references) that is explicitly attached as additional information, as well as relations or links (cf. Section 5.4).
- *Process Definitions.* Once a concrete annotation workflow has been defined by means of an arbitrary process definition language, the workflow specification file is made accessible for all participating clients. Thus, the data system includes a set of concrete workflows, which can be loaded by individual users. In addition to that, these predefined process definitions can serve as templates, that is, they can be reused and potentially edited in further annotation projects.
- *Workflow States.* As will be explained in Section 6.4.2, annotation workflows permanently undergo changes with respect to the recent state. This is to be associated with the tasks that have to be accomplished as well as with the media documents to be processed. In the case of collaboration, also activities from different participators, especially human, play an essential role. Hence, updates of (shared) annotation workflows are also stored and managed within the central server application.
- *Available Services.* The proposed conceptual framework emphasizes flexibility with respect to the integration of annotation services in order to support different forms of annotation (as digital object or as process). Among these services can be internal tools and functionalities, independent applications, or external services, for example, web services. For this purpose, annotation services are organized by indicating locations for invocation, specific APIs, or pointers to code parts. Furthermore, additional attributes are managed which enable a further classification of services, being useful in the scope of workflow control and data junction (cf. Section 6.1).
- *Configuration Data.* As revealed by the conducted analysis on multimedia annotation systems (see Chapter 3), different configurations have to be customized

in order to prepare the system for the annotation process. Besides of individual project settings such as specific user interface properties, especially commonly used configurations need to be centralized. For instance, among these common customizations are workflow definition files which determine workload distribution or the applicable annotation services.

- *User Data.* In Section 6.1, task or content distribution has been identified as one key requirement for process-based annotation systems. This presupposes that an account-based storage management of users is provided, in which accounts can also be assigned to groups associated with respective roles and access policies.

The *View* layer represents any visual component at the graphical user interface. Besides of the general elements of the user interface, single views display the available tools or services that are previously assigned to respective tasks of the annotation workflow. These views can be either own framework-specific graphical components that allow the control of external services and visualization of respective generated data, or entire external applications as components which already comprise a graphical user interface. Consequently, not every component requires the supply of a graphical user interface element.

The *Controller* layer constitutes the *Business Logic* of the framework. It comprises the functional concept of the fat-client as well as the server application within the incorporated multimedia annotation server. While the server is particularly employed for storage and management of the whole set of relevant data, the client application provides control and treatment of information, especially with regards to incoming user and system interactions. For this purpose, the client includes two different kinds of controllers: *local controllers* and one *global controller*.

Local controllers are assigned to every graphical component of the *View* and act as interfaces between component and application. As such, they realize the *local business logic* for an assigned service. In general, the effective realization of these interfaces is defined by the functions offered by each respective service, as well as by the type of information which results as output data. With respect to the provided functions, a specification is given in Section 6.3.4. General data types that can be assigned to sub-processes and tasks of annotation (and thus result from services assigned to these activities) have been described in Section 5.2.

In addition to local controllers, a central component serves as *global controller* and represents the *overall business logic* of the client application. The global controller is applied for workflow control and execution purposes, at which, among others, the interaction between different integrated system components is mediated. Here, the global controller instance does not obtain information about the concrete implementation of participating components, since this level is concerned by the incorporated local controllers. Thus, the role of a *broker* is particularly emphasized.

The intermediary role of the global controller instance is realized by the application of the *Mediator* design pattern. This pattern includes a central instance which defines the cooperation between different objects within a common framework and thus coordinates the overall behavior of the environment [GHJV95]. Figure 6.4 pictures the structure of the Mediator pattern in UML representation. Here, the global controller is regarded as instance of *ConreteMediator* and *ConcreteColleague* objects are implemented by the set of local controllers. A realization of the abstract Mediator is not intended. This is explicitly allowed if only one Mediator instance is required [GHJV95]. Thus, it can be implemented according to the *Singleton* pattern [GHJV95], which limits instantiation to one single object.

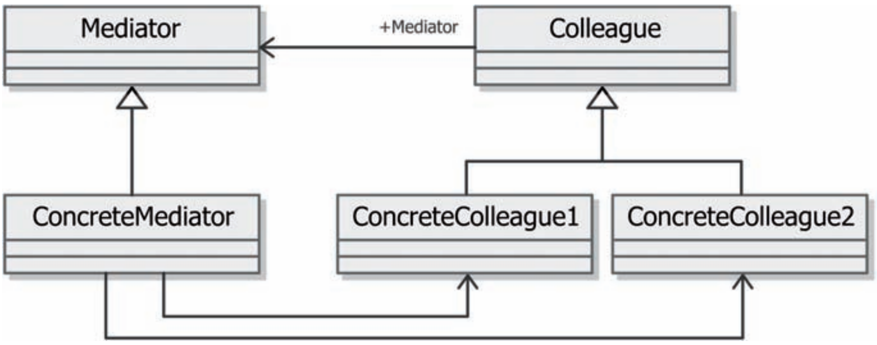


Figure 6.4: Structure of the Mediator Design Pattern. [GHJV95]

The conceptual framework implements the Mediator pattern by the establishment of a *Process Engine* component. According to the principles of the pattern, it defines the cooperation and interaction of multiple objects, and holds an intermediary role by coordinating the overall behavior of the system [GHJV95]. This is especially provided by the sub-element *Component Broker* (cf. Figure 6.2). Furthermore, by means of an additionally integrated *Workflow Handler*, workflow control can be supported with regard to transitions between workflow phases, sequences of sub-operations, passing through loops, and re-entries to other phases of the workflow. The specific processes and sequences within the annotation workflow are defined by task groups and sub operations, which can be pooled into several system components.

The global controller is key component with respect to the establishment of *Event-based Communication* and a *Service-oriented Infrastructure*, so that the specific notification strategy of a MVC model is substantiated [SN92]. An event-based system is a shared environment composed of multiple active software components and a service that realizes event transmission, commonly called Event-Broker or Event-Dispatcher [Sch00]. There, active software components may obtain the role of an event publisher or an event consumer. The Broker component appears as event transmitter and is responsible for inter-service communication and interoperability, as well as workflow control and governing. As a result, a service-oriented infrastructure can be implemented. Service-oriented infrastructures focus on the realization of components as modular services that can be invoked by certain clients. Thus, based on a service-oriented infrastructure realized by the global controller, modularized sub-processes of collaborative annotation can be individually invoked and controlled.

6.3.3 Component-based Environment

With respect to the basic structure of the reference architecture, a modularization by the incorporation of components is proposed. The entirety of components determines the specification of the characteristics and functions that need to be offered to comply with requirements for a process-based annotation system (cf. Section 6.1). This design-related decision is basically based on the requirement for pooling identified sub-processes of annotation into functional units that are mutually delimited in order to typecast task areas that can be invoked by a central control unit. Furthermore, the architecture must enable administrators to integrate, replace, and delete services that can be assigned to these task-related modules. The concept of *software components* applies to these specific requirements.

A software component is regarded as an enclosed unit that provides specific functionalities. Components are context-independent, that is, they can be embedded into a higher-level system and combined with other components. According to the principles of a *black box*, the concrete implementation of a component is concealed from its accessing instance [Szy02]. Thus, software components must provide specific interfaces which allow system-managed access, communication, and cooperation. In doing so, information is offered about the comprised functionalities as well as the demands on the entire framework [CFM$^+$00].

Specific requirements of workflow control and visualization, flexible service integration and supply, data consistency and coherent junction are considered by the integration of the key framework components *Process Engine*, *Process Visualization*, *Tool Selection*, *Communication*, and *Framework Interface*. As described above, the *Process Engine* implements the Mediator instance as global controller of the MVC model. The *Process Visualization* component is responsible for the visualization of a concrete workflow instance and the consideration of respective user interactions. As will be described in Section 6.4.2, a proposed visual-interactive concept for *Process-driven User Assistance* includes workflow-based user guidance at different levels. For this purpose, also *traditional* interaction elements for *Tool Selection* are incorporated. Moreover, the *Communication* component realizes the connection between elements of the shared framework. First, the communication between the framework's central Multimedia Annotation Server is established. Additionally, access is warranted to external information resources such as web servers or multimedia databases, external annotation services, or external annotation servers which, to give an example, might provide and store semantic annotations or ontologies. For the latter case, a *Framework Interface* component is incorporated. The same way as the concept of software components, it implements the interface for the interaction with external services or resources, for instance, by comprising specific APIs. In addition to that, *Media* components are incorporated which can be assigned to the annotated media documents.

Based on sub-processes of multimedia annotation identified in Section 5.2, the abstract components *Configuration*, *Selection*, *Addition*, *Exploration*, and *Externalization* have been conceived. The *Configuration* component is responsible for any administrative activity performed by authorized users. It provides input interfaces for several configuration options such as users and task management, process specification, or project settings. *Selection* components represent any service for content selection or segmentation. For this purpose, interfaces must be provided that support interaction with media players and data visualization components. Analogously, *Addition* components are responsible for the attachment and conjunction of any kind of annotation. *Exploration* components serve as means of reception, browsing, navigation, and searching. Basically, information is visualized, and features for manipulation of the data representation such as filtering or sorting are provided. In addition to that, specific search functionalities may be integrated. Examples are specific timeline views for continuous media like audio or video, or annotation structure visualization tools. Finally, *Externalization* tools are responsible for data processing in order to reuse the generated information in external environments or projects.

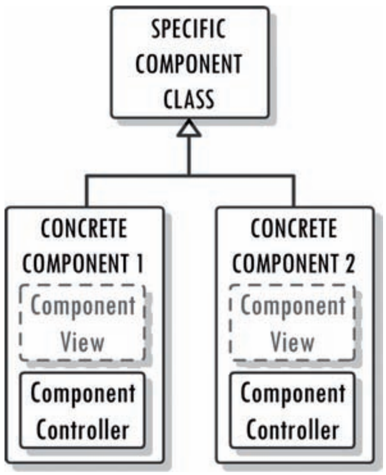


Figure 6.5: Basic Structure of Process-related Architecture Components.

The basic structure of framework components is pictured in Figure 6.5. As illustrated, components are composed of a local controller instance and, optionally, of one graphical user interface representative. As already stated, the local controller element is employed as interface between the specific component and other elements of the framework. The dispensability of an additional graphical element depends on the special function a component fulfills in the framework context. No user interface needs to be incorporated for parts of the business logic, such as the *Process Engine* or the *Framework Interface*, or in the case of invoked software components or stand-alone applications which already provide a graphical user interface by themselves. In contrast, annotation services may be associated which do not offer an user interface, but rather deliver certain data as response on specific requests. Here, the integration of a graphical interface might be required, if a manual control of the specific service and/or editing of resulting data is demanded.

Furthermore, Figure 6.5 shows that components might be implemented several times. This especially applies to all components which are related to sub-processes of annotation, so that multiple concrete tasks which belong to a certain common sub-process can be covered. For instance, two different services might need to be integrated in order to segment video documents according to detected shots or scenes. Or, to give another example, different media formats need to be supported, each requiring a different display and control element.

In the following, a more detailed specification of the different components comprised by the architecture model will be provided.

6.3.4 Characteristics of Architecture Components

In this section, the fundamental characteristics for each of the components introduced in the previous part are elucidated. Since the integration of a graphical user interface element is not necessary for all components, the described characteristics are exclusively related to the respective local controller element. Hence, the focus of this section is on the offered functions, the internal procedures, and interdependencies as well as interactions with other elements of the framework. In doing so, the overall behavior of the framework can be described. In addition to that, essential aspects for the definition of interfaces are constituted.

I. Process Engine

The Process Engine component represents the central control instance which implements the *Mediator* within the proposed adapted MVC model. The key functions can be summarized as workflow control, service enactment and coordination, and data flow management. Here, this entity can be compared to a *workflow engine*. As illustrated in Section 2.2, workflow management systems usually comprise on or more workflow engines which interpret concrete workflow instances, interacting with workflow participants (humans and machines) [CB04]. Furthermore, workflow engines also provide specific invocation capabilities to activate applications which are necessary to execute particular services [Hol95].

In order to comply with the described key functions, the Process Engine is subdivided into three types of sub-components: a *Workflow Handler* and a *Component Broker*. While the Workflow Handler is primarily responsible for workflow control, the Component Broker realizes the main functions of the Mediator model, such as component coordination and consistency-founded notification. A delimitation of both subcomponents is to be regarded as indispensable, since the graphical-interactive concept proposed in this work also includes the possibility to work without explicit workflow support, so that a *silent* execution of the workflow Handler is required (see Section 6.4.2). In this case, the Component Broker still handles relevant aspects of service coordination with respect to data junction of different annotation formats or synchronization of annotations and media documents.

I.1 Workflow Handler

The main functions of the *Workflow Handler* component refer to aspects of workflow control in general. Initially, the Workflow Handler reads the workflow definition entity previously parsed by a *Process Definition Language Interpreters* (PDLI) (cf. Figure 6.2). According to the used process definition language, a suitable PDLI interprets and transforms a concrete workflow instance definition into a uniform framework-internal structure. Subsequently, the transformed workflow instance is processed according to user-specific properties such as tasks and contents assigned to the current user, and specified roles and access rights. This light weight workflow specification is forwarded to the local controller object of the Process Visualization component, which represents the workflow in user-adapted form. According to the latter point, the Workflow Handler has direct access to the *Process Visualization* component (cf. Figure 6.2). How the workflow is actually enacted, depends on the specification included by the process definition. Here, either an initial user decision or interaction is expected, or the first task (with a respective service invocation) is automatically initiated. With respect to workflow control aspects, the key functions of the Workflow Handler can be regarded as *Workflow Event Reception* and *Workflow Event Forwarding* (see Figure 6.6).

Workflow Event Reception: Essentially, the Workflow Handler waits for the arrival of workflow-specific events, which can be divided into *Service-related Events* and *User Interaction-related* Events. This classification applies to the definition of workflow participants, which includes technological as well as human resources (see Section 2.2.1). In the scope of *Service-related Events*, the Workflow Handler is notified by the *Component Broker* when a tool or service has completed a processing unit. Subsequently, the Workflow Handler computes the next work item (task) by checking the recent process definition. Moreover, in the event that the user needs to manually choose the following step, the Workflow Handler receives an *User Interaction-related Event*. In this context, the task selection process is performed within the Process Visualization component.

Workflow Event Forwarding: Once a subsequent work item has been determined, the Workflow Handler notifies both the Process Visualization component and the Component Broker instance. In doing so, the provided workflow visualization is enabled to update the representation according to the current state. That is, the visualization indi-

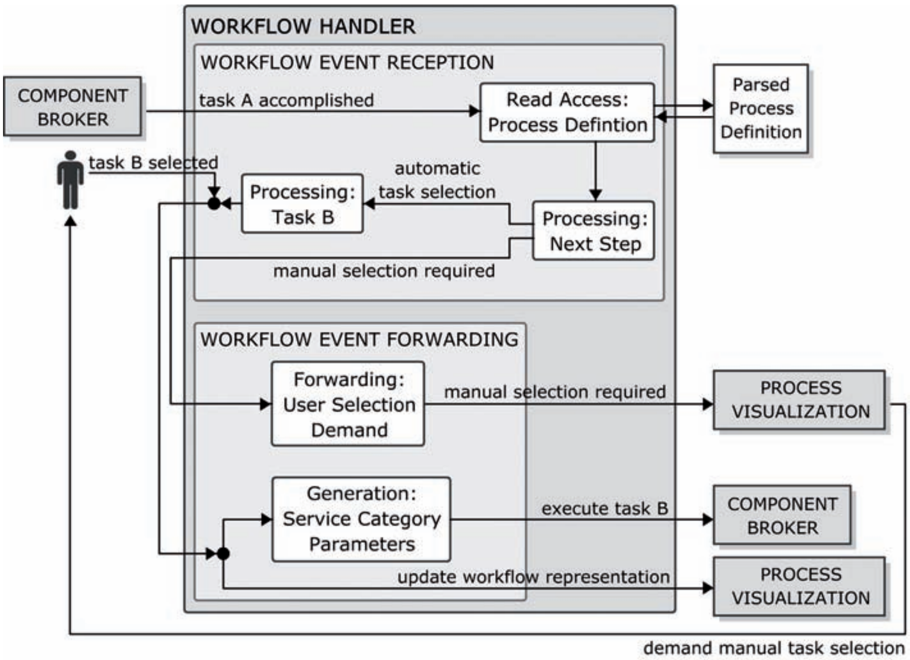


Figure 6.6: Functions and Interactions of the Workflow Handler Component.

cates for the current task or work item, as well as the tasks already accomplished. With respect to the interaction with the Component Broker, forwarding of the next task aims at enacting the appropriate service(s).

I.2 Component Broker

According to the realization of the *Mediator* pattern, the Component Broker is responsible for the management and coordination of all participating framework components. For component management purposes, a suitable design pattern is the Factory Method [GHJV95]. This method allows the instantiation and integration of formally unknown components. Furthermore, it determines an explicit location for component instantiation decision procedures, which are specified within the process definition. Besides of aspects of component coordination, the Component Broker considers data junction (see Section 6.3.5) and workload monitoring (see Section 6.4.2). As illustrated in Figure 6.7, this implies the following key functions: *Service Invocation*, *Workflow Event Forwarding*, *Data Junction*, *Modification Broadcasting*, *Workload Monitoring*, and *Synchronization*.

Service Invocation: When a specific task is to be performed, the Workflow Handler component forwards a parameterized *execute task* request, and the Component Broker determines the applicable service(s). This is based on the transfered parameters, which refer to sub-process or task-related or service categories as described in Section 6.2. Subsequently, the selected services are invoked by checking the attached location,

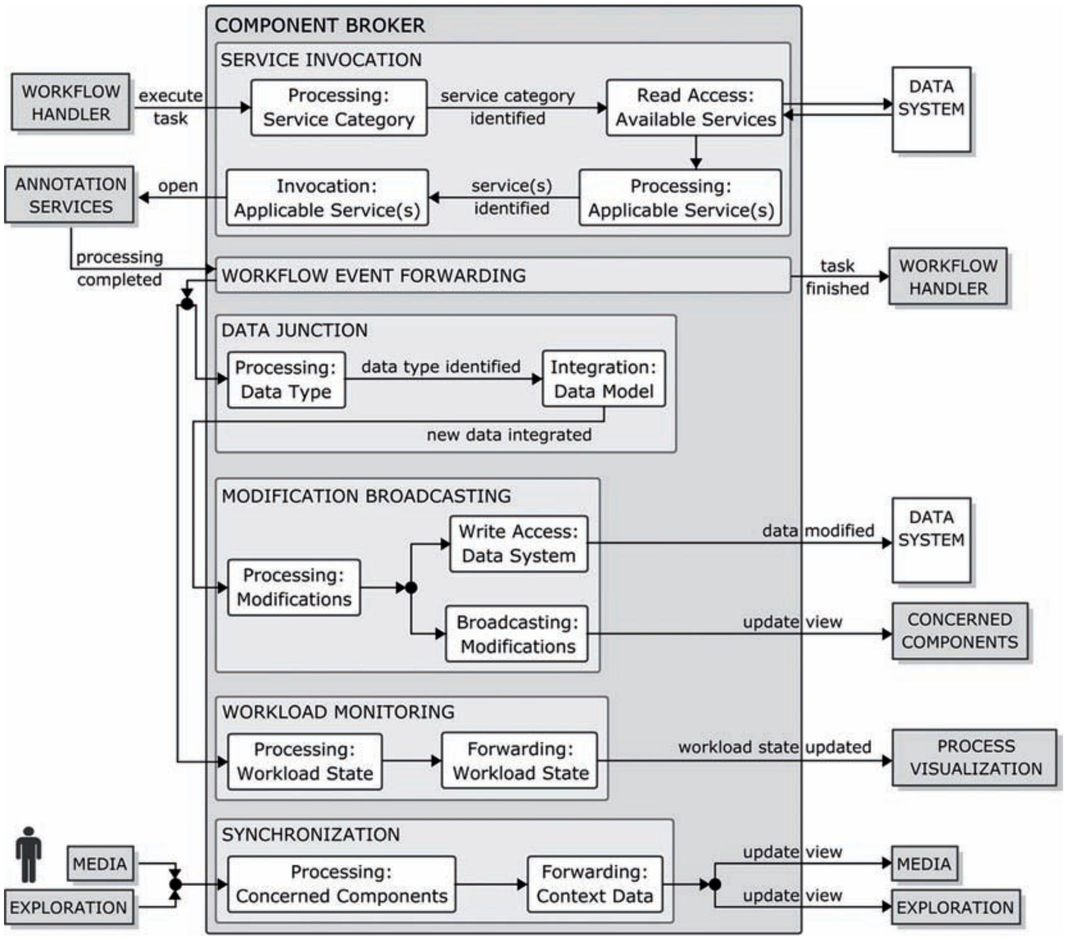


Figure 6.7: Mediator Functions of the Component Broker.

code fragment, or API specification. This forms the basis for an explicit supply of required services, i.e., user interface components which represent the provided services, depending on whether the specific service can be applied in order to accomplish the recent annotation task to be performed. Obviously, closing of not relevant services is included in this function.

Workflow Event Forwarding: Once a service has finished processing, it informs the broker instance about the completion of the respective task. The Component Broker then forwards this message to the Workflow Handler, which determines the next steps to conduct. In addition to that, procedures related to the consecutive functions are triggered.

Data Junction: Since different services are integrated in the annotation process which generate various types of annotations, this data needs to be commonly managed in a uniform manner. For this purpose, the Component Broker identifies the type of incoming data based on the category specification attached to the notifying service.

Thus, the Component Broker is enabled to integrate new information into the specific data model provided by the framework. A detailed description of the architectural structure for data junction will be given in Section 6.3.5.

Modification Broadcasting: After updating the data model, modifications are forwarded to the server application in order to establish consistency with the centralized data system. In addition tho that, modifications are also sent to all client components which visualize or process the same data at modification time. Thus, consistency is guaranteed within the client application, which is particularly relevant for synchronization.

Workload Monitoring: If multiple media objects have been previously selected for annotation, or the annotation tasks have been distributed to different users, the framework provides monitoring facilities with respect to the annotation progress. For this purpose, the Component Broker computes the tasks which a specific user account has to accomplish for each selected media object. This information is gathered from the predefined process definition entity.

Synchronization: As pointed out in Section 2.1.4, *synchronization* of the different media objects contained in a multimedia document, as well as the objects' annotations (which also can be media objects) is a relevant aspect of multimedia annotation. In general, synchronization considers the spatial, temporal, or logical relationships between involved entities. After explorative user interactions including selection of specific annotations or the display and/or control of a media object, the Component Broker informs concerned components in order to achieve respective updates. Consequently, synchronization primarily concerns *Media* and *Exploration* components.

II. Process Visualization Component

Workflow Management Systems which allow for user interactions in the context of process execution usually work with so-called *worklists* [Hol95]. Worklists are comprised of *work items* which can be regarded as the elementary pieces of work and are generated by the incorporated workflow engine(s) [LAH08]. A worklist may be visible to the user, presenting the workflow and prompting users to manually select items of work, and indicating for task completions [Hol95].

Both the *Process Engine* and the *Process Visualization* components constitute the process management system within the proposed conceptual framework. Basically, the Process Visualization component represents a concrete workflow instance, permits users interactions, and provides information about the recent workflow state (see Section 6.4.2). According to this, the main functions can be regarded as *Graphical Workflow Transfer*, *Interaction Treatment*, and *Status Update* (see Figure 6.8).

Graphical Workflow Transfer: The key function of the Process Visualization component is to transfer a workflow instance managed by the Workflow Handler into a

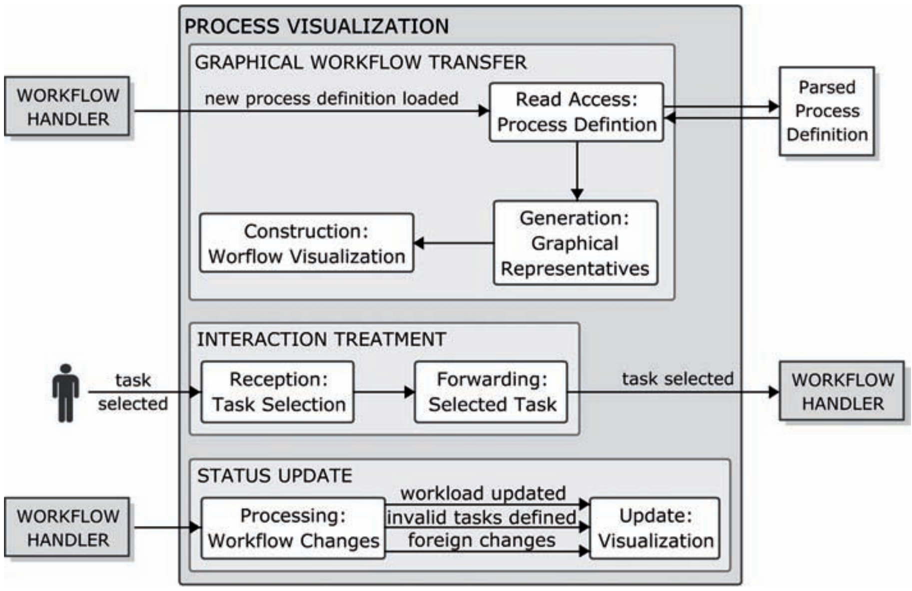


Figure 6.8: Functional structure of the Process Visualization Component.

human-interpretable graphical representation. For this purpose, the core elements of a process definition, such as activities, splits (decision nodes), joints, or conditions, are mapped to graphical base objects. After a preprocessing procedure, in which especially transitions and conditions are checked, the executive order is constructed. A detailed description of the process visualization strategy conceptualized in this thesis will be given in Section 6.4.2.

Interaction Treatment: The process management system allows user interactions referring to task selection activities. This is the case if the current user wants to switch to an arbitrary (admissible) task, or when multiple alternative tasks can be selected. The Process Visualization component accepts task selection interactions and forwards them to the *Process Engine* component which initiates service invocation.

Status Update: During a running annotation process, modifications are permanently conducted which refer to workflow-related aspects. For instance, this includes the state of a previously defined workload, that is, a number of tasks and/or a set of media objects that need to be edited. Furthermore, the visualization component needs to represent the recently processed task, for instance, by highlighting the respective graphical representative. In some cases, certain tasks can not be performed due to the current workflow state. To take an example, it is conceivable that after the selection of media objects from a database of images and videos, the workflow instance offers the alternative segmentation tasks *video shot detection* and *image region segmentation*. Thus, one of the two alternative tasks always needs to be deactivated depending on the format of the previously selected media object.

III. Tool Selection Component

As will be described more detailed in Section 6.4.2, the visual-interactive concept for *Process-driven User Assistance* proposed in this thesis realizes different levels of user guidance during the annotation workflow. Thus, besides of controlling the execution of annotation services by means of workflow management principles, also *traditional* tool selection components need to be integrated, such as menu bars or toolboxes. Here, classical *Tool Selection* components need to be based on the loaded process definition file, referring to the tools which need to be provided on the one hand, and the activation of tools during run-time on the other hand, the same way as the Process Visualization component. In contrast to Process Visualization, all relevant notifications are conducted by the Component Broker and not by the Workflow Handler (see Figure 6.9). In doing so, a deactivation of explicit workflow execution can be ensured. The main functions resulting from these considerations are *Toolset Initialization*, *Interaction Treatment*, and *Status Update*.

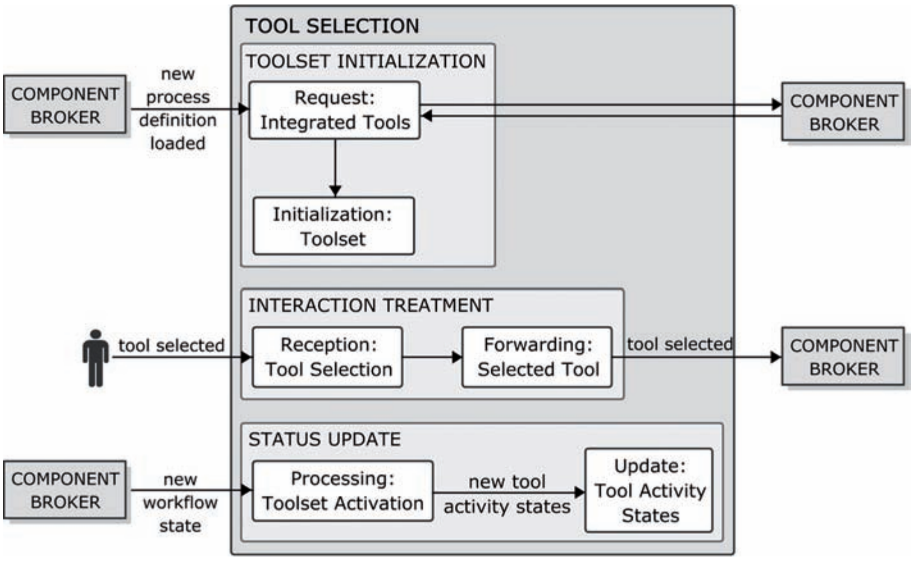


Figure 6.9: Characteristics of Tool Selection components.

Toolset Initialization: Once the Component Broker broadcasts that a new process definition has been loaded, Tool Selection components request a list of all tools (services) which are integrated in the recent workflow. Subsequently, the list of tools is constructed.

Interaction Treatment: Selection of tools is performed by human participators in manual manner. This information is forwarded to the Component Broker, which is responsible for respective tool invocation. Thus, the function *Workflow Event Forwarding* realized by the Workflow Handler is replaced. Here, instead of forwarding a task request, the required tool is named explicitly.

Status Update: As described above, different states of a running workflow might require different activity states of a tool. That is, in some cases the invocation of a specific annotation tool is not legal. Hence, the activity states for all tools need to be updated after each workflow step (which triggers a workflow state change).

IV. Configuration

The *Configuration* component is responsible for all administrative processes and configuration of the application. It provides input interfaces for the customization of general application and project properties as well as the management of user accounts, groups, roles, and access right. In addition to that, workflow- and task-related settings can be edited. For instance, distribution of tasks to different users or user groups can be stored. In general, the administration component provides interfaces for different types of information entry, and allocates data to its proper destination. The resulting basic functions of the administration component are *Extraction* and *Integration*.

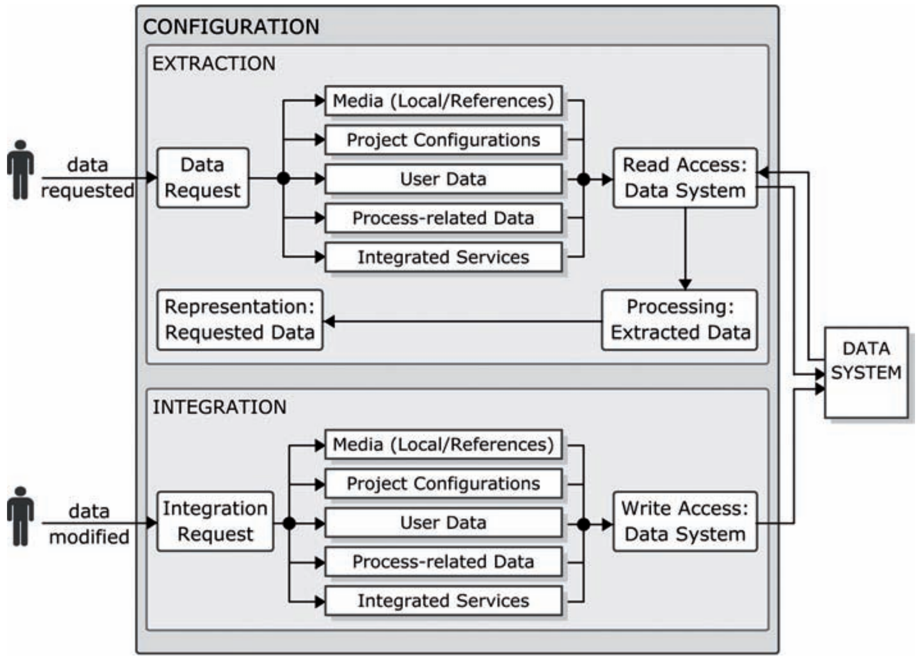


Figure 6.10: Administrative Functions of the Configuration Component.

Extraction: In order to enable the creation, editing, and removing of administrative or configuration data, the component provides read access to the data layer, so that the required information can be displayed on the graphical user interface.

Integration: As illustrated in Figure 6.10, the Configuration component implements the write access to the data system, depending on the detected type of information.

V. Media

In general, *Media* components are incorporated in order to display the media objects processed within the annotation framework. Depending on the given media format, specific functionalities have to be provided which support navigation and media control. Furthermore, additionally features may be required with respect to the display and interaction with existing anchors or annotation validity areas (cf. Section 2.1.4). Interaction with such areas primarily include manual selection, but may also refer to creation and editing procedures, i.e., also manual segmentation can be supported. As pictured in Figure 6.11, the basic functions of Media components are *Display and Playback*, *Navigation and Control*, and *Anchor-level Interaction*.

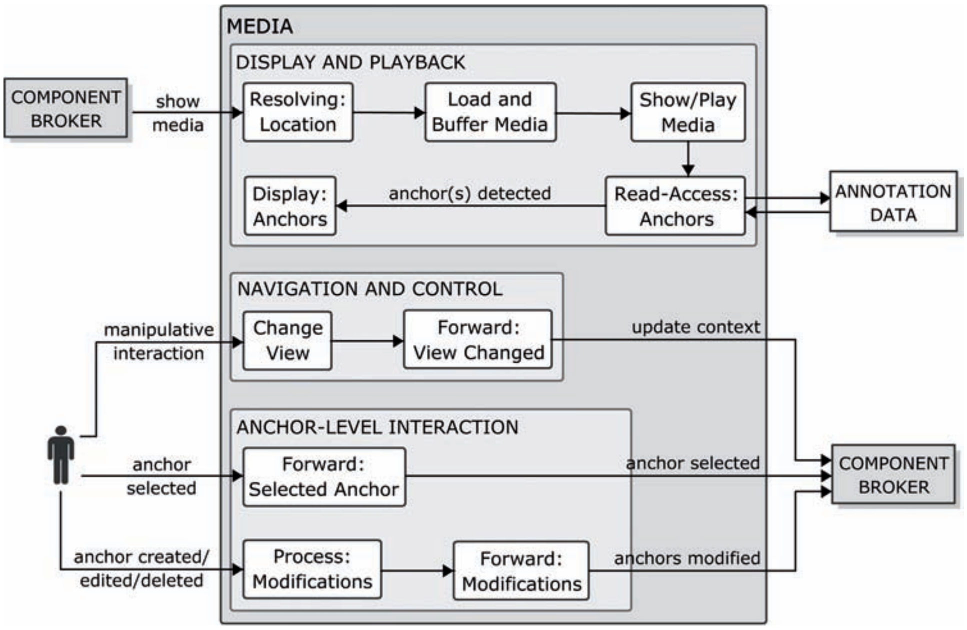


Figure 6.11: Media Component Functions.

Display and Playback: Media components provide specific display areas which are adapted in conformity with the supported media type. For media display purposes, the location of the object to display is delivered by the *Component Broker* instance at invocation time. Depending on the form of location indication, specific preload or buffering functionalities might need to be involved, e.g., if streaming is required. Additionally, some of the annotation systems examined in Chapter 3 reveal that specific areas are shown and highlighted upon the respective media display area, for instance, video or image region segments, or specific sub-elements within a 3D-model. Thus, also anchors, i.e., specific annotation validity areas, might need to be displayed.

Navigation and Control: Playback elements are required for continuous media like audio or video, or zoom and panning functionalities for images and web documents like PDF files. Any navigation and control event is sent to the *Component Broker* in order to guarantee synchronization. In doing so, data visualization components which display annotations attached to the recent media object can be informed that the context has changed, and that another set of annotations probably needs to be highlighted.

Anchor-level Interaction: Anchor-related interaction particularly refers to the manual selection of anchors which may concern an entire media object or parts of it. The *Component Broker* has to be notified in order to inform other components that a specific area has been selected and respective annotations need to be highlighted. In addition to that, especially video and image-based annotation systems show that generation and editing of segments which refer to a specific region is to be enabled upon the media display area. Such forms of segment interactions also need to be delivered to the central control instance in order to update the data system and further system components.

VI. Selection

The general objective of *Selection* components is the generation of anchors, that is, the selection of areas which serve as point of attachment within a media document. For this purpose, different manual or automatic services can be implied (cf. Section 5.2.4). A relevant aspect is the interaction with other system components. Besides of working with an independent tool or running a single service, area selection may be performed upon *Media* components as described in the previous paragraph, or within *Exploration* components. For instance, Kipp [Kip08] points out that video annotation (time-based selection and information attachment) is mostly performed in the context of working with timeline-based representations. As a result, basic functions of Selection components, *Media Location Forwarding* and *Anchor Forwarding*, focus on the information exchange with the Component Broker. In addition to that, especially for the case of the application of external automatic services, additional graphical user interfaces might need to be provided in order to allow a manual interactive manipulation of computed data, so that potential errors or misinterpretations can subsequently be edited. Thus, as shown in Figure 6.12, also *Interactive Manipulation* is (optionally) offered.

Media Location Forwarding: On service invocation time, the *Component Broker* delivers the location of the media object to be processed by transferring a URL, URI or similar forms of location indication. Thus, the applied service can access the respective file(s) in unassisted manner.

Interactive Manipulation: As stated above, specific interaction elements might need to be provided in order to allow a manual interactive manipulation of computed data. This is an optional function that especially fits to the application of automatic services or to subsequently edit already existing anchors. For that purpose, specific interfaces must be realized that are able to process structural metadata, such as coordinates, time-based data or standardized formats like X-Pointers or Image Maps (see Section 5.2.4). The concrete implementation depends both on the supported media format and the type of structural metadata which is handled.

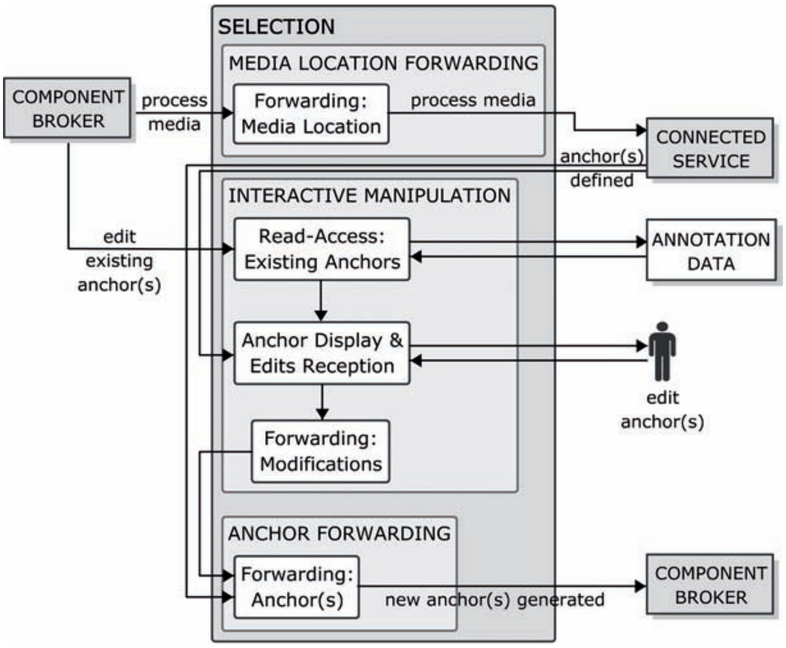


Figure 6.12: Functions of Selection-related Components.

Anchor Forwarding: Once selective annotations have been generated and possibly manually edited, the new data is forwarded to the Component Broker in order to integrate new information into the given data model and initiate data consistency procedures.

VII. Addition

Concrete implementations of the *Addition* component enable supply, representation and editing of annotated data such as descriptive metadata, categorization, commentary, further multimedia content, etc. Consequently, a relevant aspect is the consideration of different forms of information, which have to be supported by providing suitable input interfaces. Assuming that the *Component Broker* has already registered the media anchor(s) which are to be be connected with new information (e.g., through previous selection of an existing media segment), the basic functions of Addition components are *Information Generation*, *Interactive Manipulation* and *Information Forwarding* (see Figure 6.3.4).

Information Generation: For annotation generation purposes, interfaces are applied which allow the communication with tools and services for information creation and editing. Among these are both manual and automatic services. With respect to the latter case, respective interaction elements might need to be provided on the graphical user interface for subsequent editing purposes, comparable to *Selection* components. A representative example are specific metadata extraction services.

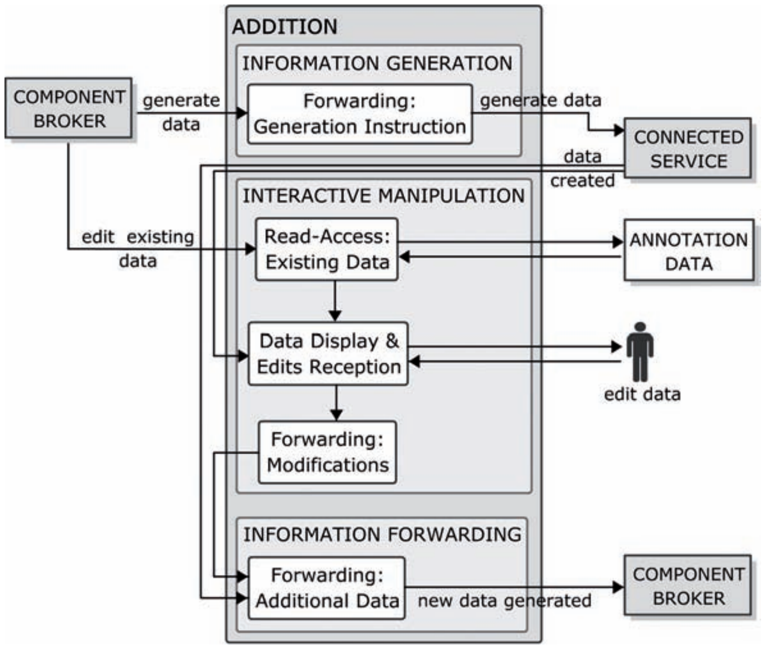


Figure 6.13: Functional structure of Addition Components.

Interactive Manipulation: As with Selection components, manual creation and modification of data might need to be supported in the case of automatic information generation. For this purpose, the graphical user interface must offer graphical and interactive elements for information representation and manipulation. The concrete implementation strongly depends on the supported data format.

Information Forwarding: Once the new additional information has been created (or edited) , the data is forwarded to the Component Broker, at which the correct association and integration into the data model is performed (see Section 6.3.5).

VIII. Exploration Component

In general, components which apply to Exploration procedures include functions for searching, viewing, and navigation on the given media and annotation data. In addition to that, means of manipulation of the data representation are provided which can be useful when working with larger data sets. As pictured in Figure 6.14, the functional area is divided into features for *Extraction*, *Data Representation*, and *Interactive Manipulation*.

Analogous to other framework components, external services can be associated, for instance, in order to integrate specific visualizations or search engines. In accordance with the concept for *Process-based User Assistance* elucidated in Section 6.4, at which *permanent* UI components will be specified, at least one fix component is provided that aggregates the complete information space evolved within the running annotation process. Hence, this component is enabled to represent information in cooperation with any invoked service at run-time. This is warranted by the specific synchronization and notification procedures provided by the *Component Broker* object.

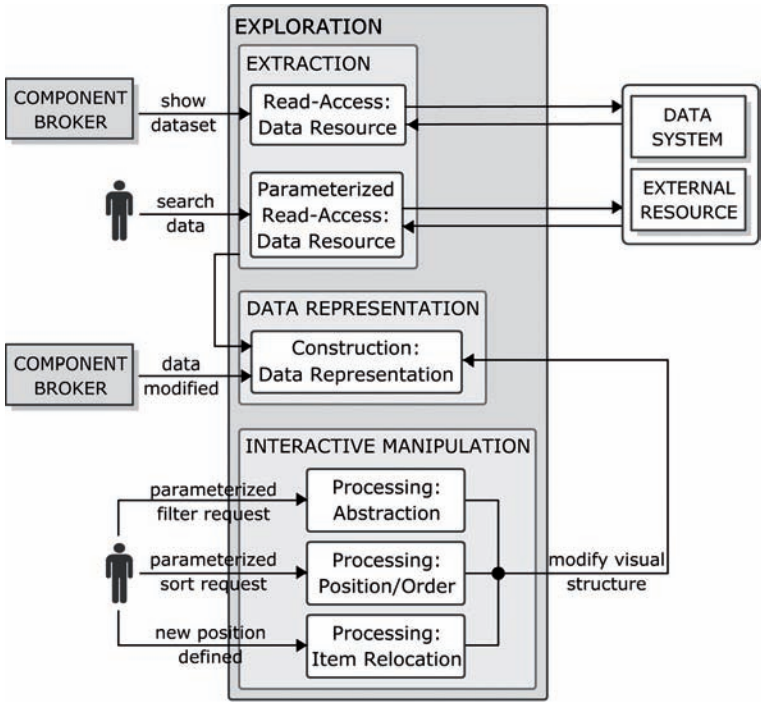


Figure 6.14: Functional Structure of Exploration Components.

Extraction: An Exploration component realizes read access to specific data resources in order to represent this information synchronized with the respecting media objects. Here, data resources include the framework's data system, but also external resources such as data of further projects or external annotation servers (at which communication is supported by means of the *Framework Interface* component). Moreover, extraction includes the processing of search queries, which can be regarded as specifically parameterized extraction requests.

Data Representation: Data which results from a general extraction request or a search query is represented on the graphical user interface. Depending on the type of acquired information, various forms of representation are provided from simple lists to more complex visualization structures. Basically, among different types of information are media documents and objects, media subsets and segments, and attached additional information. Furthermore, a current data representation is updated when the *Component Broker* sends a notification of data modification.

Interactive Manipulation: Particularly if users are confronted with larger sets of information (including search results), which also may dynamically expand in the course of collaboration, a manual manipulation of visualized data is necessary in order to ensure orientation within the given information space [Sch07] and cognitive load reduction [SFZ06]. Examples of respective technical auxiliaries are *filtering* and *sorting*[1] according to specific parameters, or item *relocation* (cf. ref). The latter case can be assisted by *direct object manipulation* principles [Shn97].

IX. Externalization

At the end of an annotation process, the gathered data is externalized in order to publish, summarize, or export it. Thus, *Externalization* components are exploited in order to reuse information in further contexts, create result reports, or for further processing with other applications. Here, all relevant data might be included, such as media documents and objects, media subsets, and annotations. This results in three main functions: *Data Selection*, *Parsing*, and *Data Forwarding*.

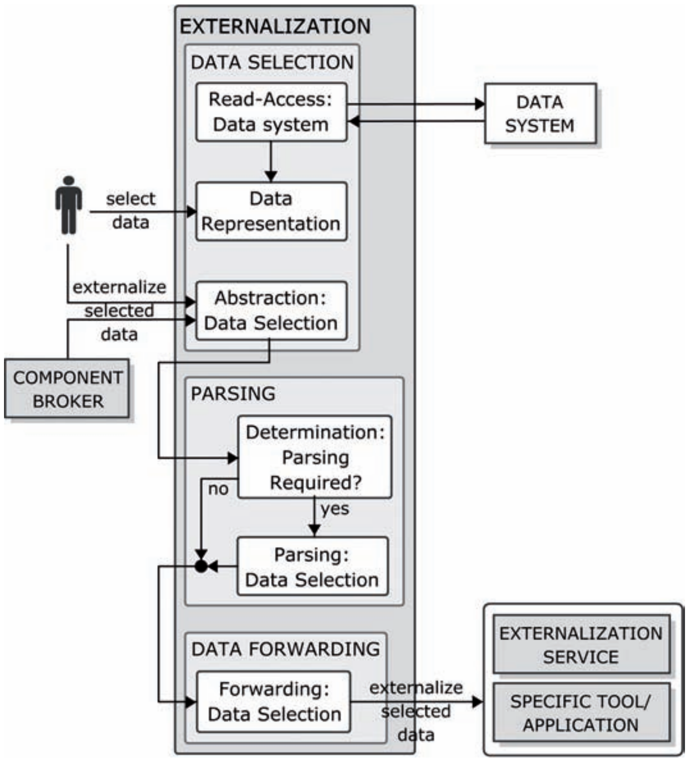


Figure 6.15: Functions and Internal Procedures of Externalization components.

Data Selection: Before passing the project's data for externalization, required parts of the entire dataset might need to be selected. For this purpose, necessary functions are data representation and manipulation, which can be provided by the component itself or realized in cooperation with an *Exploration* component (see Figure 6.15). Examples are summaries of interesting video scenes, or the transfer of a certain (annotated)

[1]In this context, the term *sorting* also includes acts of grouping, clustering, and hierarchization.

category for further statistical computing. As a matter of course, also the entire dataset can be handed over, e.g., in order to generate comprehensive reports.

Parsing: In the case of forwarding of data to other applications or services, certain specific formats might be expected. Consequently, parsing must be provided in order to transform selected data in a requested structural format. If the used externalization component is part of the implemented system, probably no previous parsing procedures are required.

Data Forwarding: Finally, the component forwards selected and potentially transformed data to its destination point. This procedure might be assisted by the *Communcation* and *Framework Interface* component.

X. Communication Component

This component is responsible for communication, transaction, and data exchange between clients and the Multimedia Annotation Server (and the centralized data system). Thus, any kind of read and write procedures belonging to the remaining system components is conducted via the *Communication* component. In addition to that, in conjunction with the Framework Interface, communication with external resources can be established. Among such external resources are (i) annotation services which can for instance be implemented as web services, (ii) multimedia databases containing "raw" data, and (iii) external annotation servers which provide sets of already annotated data. An essential aspect covered by the Communication component is the integration and management of different exploited protocols which form the base of communication. This is realized by the integration of respective *marshallers* and *unmarshallers* (cf Figure 6.2). According to these aspects, *Connection Establishment*, *Marshalling*, and *Data Forwarding* can be regarded as main functions.

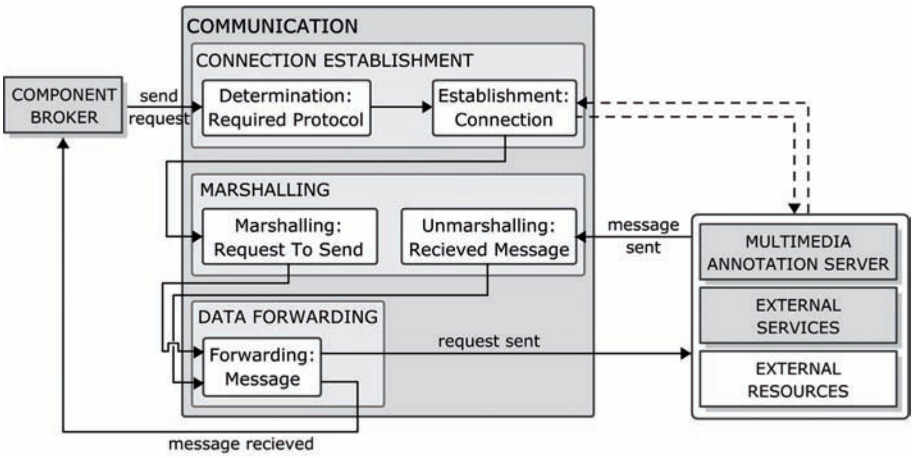


Figure 6.16: Communication-related Functions.

Connection Establishment: The Communcation component implements the connection between the different integrated components over a net (e.g., the WWW). To this end, different strategies can be employed such as constant point-to-point connection or message-based broadcasting.

Marshalling: The component provides integration and management of different modules which are exploited to (i) transform outgoing messages into a required format according to a certain protocol, and (ii) re-transform incoming messages into a framework-interpretable structure.

Data Forwarding: The Communication component is to be regarded as the interface between the *Component Broker* (including its connected client components) and the *Multimedia Annotation Server*, as well as further external services and resources. Thus, messages are transferred between these different entities. To give a better understanding, Figure 6.16 distinguishes between *requests* conducted by the Component Broker and *messages* sent by the server application or external objects.

XI. Framework Interface

In the context of this work, also service-oriented infrastructures need to be considered, since not only stand-alone software components or applications are taken into consideration, but also specific services which deliver different kinds of attachable information. On these conditions, in order to manage the annotation workflow execution within a service-oriented architecture, potentially distributed resources need to be orchestrated [GKK+08]. In this context, the *Framework Interface* component may construct and/or offer interfaces for all integrated framework components which control or communicate with external services. Hence, these interfaces correspond to the specific interfaces and features provided by the respective services. On the other side, the Framework Interface realizes an API for the entire framework, indicating for the services provided by the annotation environment. This is especially relevant in use cases, at which the framework is part of a superior process and/or environment, or in the course of interconnection with other (also heterogeneous) workflows. The Framework Interface is located on behalf of the Fat-Client, so that needless communication threads such as client-server-service (and back) are reduced.

Since the concrete implementation of interfaces strongly depends on the type of services which are to be supported, no further specification of this component is given at this point. Further related work has been treated by Heinzl et al. [HSJ+09].

6.3.5 Architectural Structure for Data Junction

As already pointed out in Section 6.1, *Data Junction* is a specific challenge in the case of multimedia annotation workflows. This can be attributed to the involvement of different media formats and their special properties on the one hand, and to the multitude of heterogeneous types of annotations which are generated during the annotation process on the other hand (cf. Section 5.2.5). Such data formats are summarized in Figure 5.8, Section 5.4. Consequently, the main challenge is a consistent and uniform storage and management of heterogeneous data in one common model.

For this purpose, a respective data model will be established in I., which is able to aggregate different forms of media formats and annotations. Here, this will be performed on an abstract level. Deepening approaches for Multimedia Annotation and Hypermedia Data Models can be localized at Hardman et al. [HBR93, HBvR94], Tochtermann [Toc95], or Westbomke and Dittrich [WD02]. The described model basically refers to the media formats and annotation types involved in sub-processes or tasks of annotation. Accordingly, the process-related components Media, Selection, Addition, Exploration, and Externalization described in the previous section will be exclusively considered in II.

I. Abstract Data Model for Multimedia Annotation

The abstract data model for multimedia annotation is based on Section 2.1.4, at which essential aspects of multimedia have been elucidated with respect to annotation validity areas (and respective structures of media objects), the basics of anchoring, and annotation of various media formats. In Figure 6.17, the presented model is pictured in an UML-oriented sketch. In the following, the basic data model entities will be explained.

Multimedia Container: *Multimedia Containers* represent the media documents which are processed within the framework. They contain at least one *Node*. In the case of composed documents such as websites or PDF files, several *Media Object Nodes* can be included. Furthermore, all *Anchor Links* which connect anchors of different nodes are managed. Hence, Multimedia Containers comprise all relevant data with respect to annotated media documents: media objects, annotations, and relations between them.

Node: *Media Objects* as well as *Additional Information* are basically managed in forms of *Nodes*. In this manner, both forms of annotation data can be associated with anchors. In addition to that, it has to be considered that annotations can also be media objects.

Anchor: Each Media Object or Additional Information element obtains one or more *Anchors*. Anchors are to be regarded as connection points for all kinds of Nodes. Here, the different types of Anchors establish specific forms of annotation validity areas (or media subset areas). As described in Section 5.2.4, among these are entire documents, spatial areas, temporal areas, or spatiotemporal areas. Thus, Anchors are defined by some sort of structural information, and can also be regarded as one form of annotation. A special anchor type is *Generic Anchor*, which refers to the *entire* document it is defined for.

Anchor Link.: The connection between Anchors defined for specific Node entities is realized by *Anchor Links*. In general, different forms can be distinguished, such as *directional*, *bidirectional*, or *multiple* links.

Media Object: *Media Objects* represent the actual media content in its different formats. Thus, concrete instance contain an indication for the location of the physical media file. Furthermore, media-specific attributes are incorporated, such as duration, size, frame rate, color density, etc.

Additional Information: Besides of Anchors as structural information, Additional Information are the second form of data entities which are regarded as annotations. As illustrated in Section 5.2.5, different types of additional information include descriptive or semantic metadata, media content, dialog artifacts, or relational descriptors.

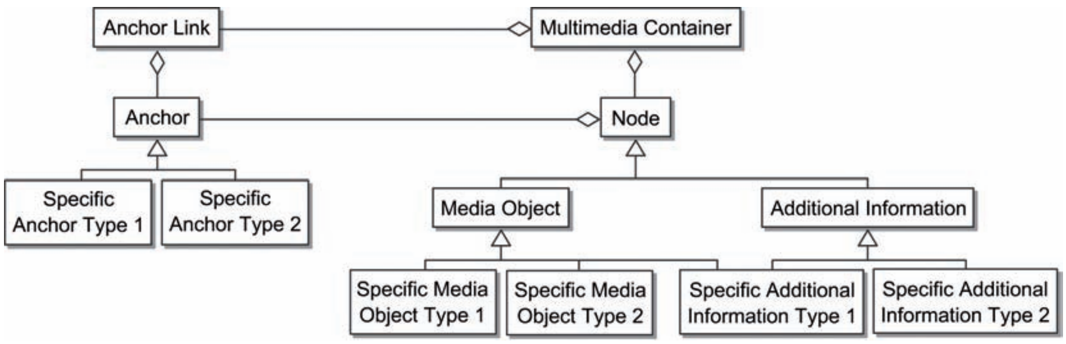


Figure 6.17: Abstract Data Model of Multimedia Annotation.

II. Structural Component Organization for Data Junction

For junction of heterogeneous data processed in the scope of annotation processes, the definition of different types of controllers on the part of the framework business logic is employed. Figure 6.18 shows the architectural structure for uniform data storage and management. Central elements are specific modules included by the Component Broker instance (cf. Figure 6.2 in the first section of this chapter), which obtain read and write access to specific entities of the common data model. Additionally, the various local controllers, which constitute the required interfaces between integrated annotation services and the entire framework, are assigned to one or more read and write modules of the broker. In doing so, required parts of the data model are associated to the internal service representatives, serving as respective input and output.

As a result, the following basic associations can be made between types of process-based architecture components and specific parts of the data model (see Table 6.2). Media components basically ensure display, control, and navigation of integrated media objects. In addition to that, it is potentially required to enable representation and modification of anchors. Selection components are responsible for the generation and

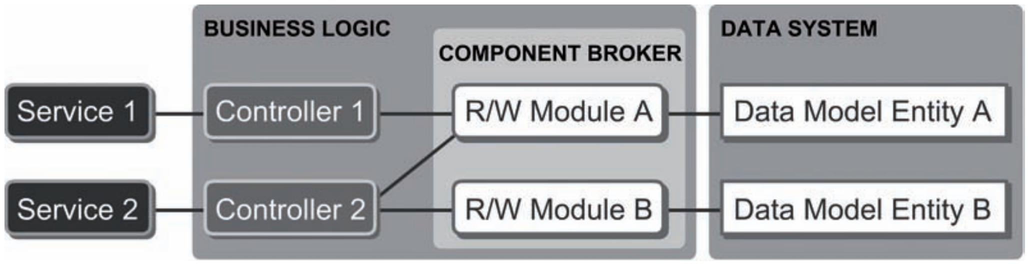


Figure 6.18: Architectural Structure for Data Junction.

editing of structural metadata which define validity areas of annotations. Thus, access on anchor instances needs to be provided. Addition components realize the creation and processing of additional information of different formats. Furthermore, the entire set of media documents and annotations including their relations is to be provided by Exploration components in order to perform search requests, general viewing, or manipulation. Hence, these components obtain access to all instances of Multimedia Container. The same applies to Externalization components, which are means of exporting or publishing the entire data set or parts of it.

Components	Data Model Objects
Media	Media Object, Additional Information
Selection	Anchor
Addition	Additional Information
Exploration	Multimedia Container
Externalization	Multimedia Container

Table 6.2: Assignment of Process-related Architecture Components to parts of the Data Model.

III. Discussion

The presented approach provides a solution concept with respect to the storage and management of heterogeneous incoming and outgoing data. It is described on a structural level, defining the relations between elements of the system architecture and the relevant data model entities. Nevertheless, no statements are given about concrete assignment strategies for these different items, leading to two problem factors. First of all, situations have to be considered in which existing annotation data is recalled, e.g., for re-editing purposes. Problems arise from the fact that requested data needs to be represented by the correct UI component (usually the UI representative of the service by which the data has been generated). To give one example, a workflow may include two different services which generate the same type of information, for instance, a video event detection and a scene detection service which produce time intervals. In this case, a later recognition of the respective UI component just based on the given data entities is not trivial. Second, in the scope of flexible and extensible systems, it must be possible to replace services, for example by new versions providing improved algorithms. As well, a solution for correct service recognition and assignment is required.

Related problem areas are addressed by research work conducted in the field of *service description* [LADM08]. Thus, approaches can be references that are associated, among others, to *service composition* or *semantic service modeling*. For instance, a semantic annotation of services can foster service discovery as well as their composition into workflows [BEP$^+$08, MK10]. To sum up, open issues have to be adhered at this point. Accordingly, these topics are discussed in Chapter 9, giving prospects for relevant future work.

6.3.6 Summary

In this section, a *Reference Architecture Model for Process-based Multimedia Annotation Systems* has been proposed. Here, the core aspects can be summarized as a Client-Server Model which is additionally structured through an adapted combination of a the *Model-View-Controller* (MVC) and *Mediator* patterns. Furthermore, in order to foster flexibility and to align to the process model of multimedia annotation defined in Chapter 5, the reference model is defined as *Component-based Environment*. In the following, the fundamental characteristics of the reference model are summarized, considering the compliance of process-related requirements which have been defined in Section 6.1.

Since one key requirement is the realization of interconnectedness between participating peers and shared data storage and management, a Client-Server Architecture has been proposed in Section 6.3.1. In this manner, the effective process execution is separated from data management, so that replaceability on the levels of both client and server is achieved. The *Multimedia Annotation Server* is responsible for a centralized storage and management of all relevant project data, and provides additionally services such as author authentication as well as accounts, roles, and rights management. Client applications are implemented as *Fat-Client*, so that a significant part of the overall functional area is located at the local computing device. Among client-sided functions are the invocation and display of graphical components on the user interface, the interaction with external services, workflow visualization and control, as well as data control and junction at run-time.

The Client-Server Model is additionally structured by an adapted combination of the *MVC* and *Mediator* architectural patterns, which have been described in Section 6.3.2. The applied MVC model separates the entire system into three layers: the *View* comprises all graphical components, the system business logic is located at the *Controller* layer, and the *Model* represents the accessed data system which is contained in the incorporated server machine. Here, the data system includes *media files* (local or referenced), *annotation data* (selections and additional information), user-specified *process definitions*, metainformation about *available services*, general *configuration data*, and account-based *user data*. The fundamental elements within the Controller layer are multiple *local controllers* and one *global controller*. Local controllers are assigned to the graphical components that represent tools and services displayed on the user interface. These controllers realize the functional concept of a component and serve as interface between the component and the global controller instance. The global controller object implements the core functions of the business logic: workflow management, coordination of components (through their local controllers), synchronization, and client-sided consistent data junction. For this purpose, the Mediator pattern is proposed, at which a central instance is defined that coordinates the interaction between different objects within the common environment, defining the overall behavior. This is achieved by providing a specific *event-based notification strategy*. Thus, it forms the basis for the establishment of a service-oriented infrastructure.

The single framework elements have been specified as *components*, which can be regarded as functional units with respect to different areas of the annotation process (Section 6.3.3). Here, the first type of components realizes the functional concept of the framework and applies to general requirements for process-based multimedia annotation systems. Thus, with except of the *Process Visualization* component, they can be located at the business logic layer. The second class of components refers to the annotation services or tools invoked at run-time. Consequently, they relate to sub-processes of annotation which have been presented in Section 5.2. Basically, they comprise a local controller and a view element which is displayed on the graphical user interface. Here, the view element is dispensable if the assigned service already offers an individual user interface, so that the component only serves as interface between service and framework. Thus, flexible service integration is fostered at different levels, that is, it is facilitated to connect specified tasks either with tools already provided by the annotation system, external services, or stand-alone applications.

In Section 6.3.4, the single framework components have been illustrated on a model basis. In doing so, the provided core functions, internal procedures, as well as interdependencies and interactions with other framework components and elements have been emphasized. In this manner, interfaces that need to be provided by all architecture components can be derived. Hence, the presented specifications must be primarily regarded as construction plan in the scope of an effective implementation of a process-based multimedia annotation system.

Finally, Section 6.3.5 provides an architectural solution concept in order to realize a uniform and common storage and management of heterogeneous data that is generated probably around one single media document. Here, specific problem factors have been discussed with respect to an adequate description of services, so that open issues can be derived concerning relevant future research work.

To sum up, this section elucidated the structural organization and behavior of the proposed system architecture and incorporated components. Here, especially the management of relevant data, synchronization and coordination of different services, and control of the entire process have been emphasized. Thus, an architectural model has been established that forms the basis for a visual-interactive solution concept. Accordingly, a concept for *Process-driven User Assistance* will be introduced in the following section.

6.4 Process-driven User Assistance

In this section, a visual-interactive concept for *Process-driven User Assistance* is presented, which aims to support users during the operation of an annotation system in the course of a concrete workflow. Since this thesis especially focuses on user-specific problems concerning a disorientation with respect to the set of tasks, the current workflow state and task to perform, as well as associated services involved in an entire annotation process (cf. 1.1), this section has to be regarded as the central contribution of this thesis. These user-specific challenges refer to the following information, which human participants need to obtain in the scope of user-centered visual process support:

- Which tasks are to be accomplished in general, and in which sequential arrangement are they to be executed?
- What is the current state or actual position within the workflow and, accordingly, which task has to be processed now?
- Which tools are available that comply with the current task?
- Which tasks have already been accomplished and which are the work items that lay ahead?

According to these considerations, the presented concept can be subdivided it into different subordinated aspects which will be elucidated in the next sections. The remainder is organized as follows. First, a basic visual design concept for graphical user interfaces will be described in Section 6.4.1 which divides the UI into different functional units. In Section, 6.4.2 a core visual instance will be proposed which allows users to obtain an overview and keep track of the entire annotation workflow, and to exert influence in an interactive manner. The basic aspects of workflow-driven service supply will be explained in Section 6.4.3. Section 6.4.4 summarizes the single aspects included in this part of the entire solution concept.

6.4.1 Basic Visual Design

Initially, a basic visual design for the UI of a process-based annotation environment needs to provided. In this context, the following aspects are to be considered which allow the derivation of general functions a specific graphical UI needs to treat in separate manner.

1. First of all, different types of user expertise referring to the operation of the annotation system as well as varying usage styles and preferences need to be considered. Thus, workflow-based assistance needs to be provided on different levels from fully workflow-driven to completely devoid of workflow support.
2. Second, the predefined concrete workflow instance has to be visualized including all workflow elements which are to be regarded as relevant for human participants.
3. Third, tools or services need to be considered which are permanently demanded during an entire annotation run (e.g., for one media document).
4. Finally, when a certain task has been selected, whether by a user or by the process engine in an automatic way, the adequate tools or services need to be invoked and represented on the UI.

Based on these four aspects, a *4 View Model* is proposed which divides the graphical UI into various areas, enclosing different functional clusters. By this means, the user is provided with unique basic element categories, at which the straightforwardness and orientation within the user-system interface is fostered [Fin05]. In this context, Khazaeli [Kha05] highlights the benefit of the provision of different access trails with respect to the usability of an interactive system. This applies to principles of human attention, especially to the *law of closeness* [Kha05]. An implementation of the view model within a concrete environment depends on various factors, at which the crucial criterion is the applied interaction device. Obviously, different graphical representations are required for mobile devices and big sized screens.

As pictured in Figure 6.19, these enclosed user interface views include a *Traditional Tool Selection View*, an *Interactive Workflow Visualization View*, a *Permanent Services View*, and a *Transient Services View*. In the following, the different views of the *4 View Model* will be elucidated.

Traditional Tool Selection View

According to different user experience with respect to system operation as well as different usage styles and preferences, the framework considers cases in which explicit visual-interactive workflow support is not required in complete extension. This might

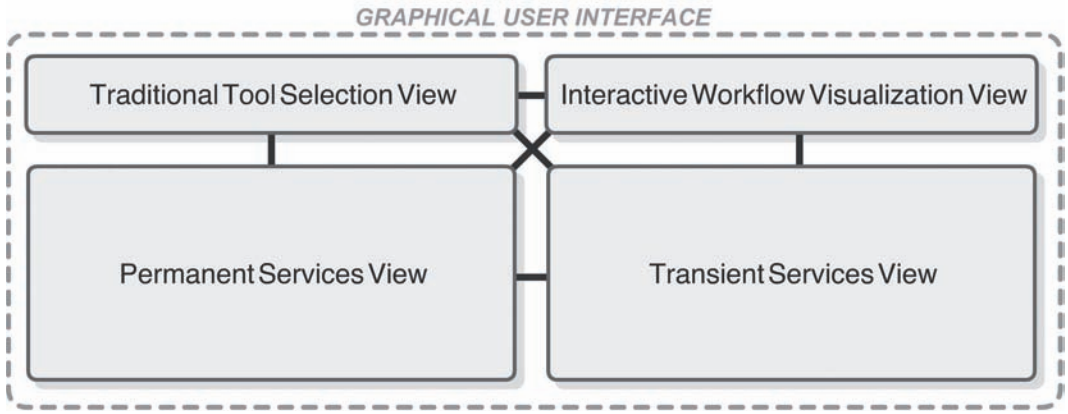


Figure 6.19: A 4 View Model which determines the Conceptual Visual Organization of the User Interface.

be the case if users do not require any workflow support, or if they want to select tools or perform tasks apart from the actual demand of the running workflow. In doing so, an excessive guidance of operators and respective negative effects are avoided. For this purpose, also traditional interaction elements have to be incorporated, whose primary function is the selection of available functionalities. Among these traditional interaction elements are menu bars which provide textual tool selection areas in prestructured manner, as well as toolboxes which provide iconized fields as representatives for respective tools and functionalities. In addition to that, as revealed by the conducted analysis of annotation systems in Chapter 3 (but also supported by many other forms of applications), specific keyboard shortcuts can be applied for tool selection activities, frequently indicated in a menu bar. In general, a key requirement in the specific context of this conceptual framework is the flexibility of such elements, concerning different tools that can be integrated into the common environment.

Interactive Workflow Visualization View

The visualization of predefined workflow instances basically includes the incorporated tasks as well as their executive sequences. It primarily aims at providing users with an overview of the annotation tasks to perform. This occurs in a personalized manner, considering different potential user roles and workloads. Interaction particularly refers to the selection of represented task items. Here, two different forms can be distinguished: human interactions demanded by the system when an interrupting point is reached in which a user decision is required, or selective interactions at which a user decides to perform another tasks than the actual activity scheduled by the process engine. As already stated, a further relevant requirement is the deactivation of the workflow visualization component, at which the automatic workflow execution is transfered to a *silent* mode. The proposed model for Interactive Workflow Visualization is elucidated more detailed in Section 6.4.2.

Permanent Services View

Tools can be identified which are potentially required and used during an entire annotation run. Among these are especially display components which contain the edited media document(s) and exploration-related tools such as visualizations for the set of generated annotations. With this in mind, a specific placement area is introduced which integrates tools or services that permanently remain superimposed on the graphical user interface. This area needs to provide position space for multiple graphical elements. Thus, different videos might need to be inspected, which show the same scene from different perspectives, or, for explorative purposes, different search and visualization components need to be integrated at the same time.

Transient Services View

The presented concept for *Process-driven User Assistance* also includes that the environment automatically supplies tools or services, so that user is provided with the correct tool(s) with respect to the task that actually has to be accomplished. In the *Transient Services View*, such kind of tools (as graphical representative of services) are inserted and (visually) removed. Such as specified for the *Permanent Services View*, it has to be possible to integrate multiple components. To give an example, the workflow definition might intend to accompany media segmentation as well as annotation attachment with a specific commentary tool, so that all conducted activities can be described by the executing user, and co-annotators are able to comprehend modifications.

In this section, a basic model has been presented which structures the UI into four independently accessible areas. These areas enclose various tools and graphical components which can be assigned to the functional fields of traditional tool selection, workflow visualization, permanent tools, and workflow-dependent transient tool. In the following section, the relevant aspects of *Interactive Workflow Visualization* will be elucidated.

6.4.2 Interactive Workflow Visualization

In general, a workflow is composed of tasks which are ordered on different structural levels. How to manage these tasks is the central issue for the execution and completion of the entire process [YLS+04]. In this context, visualizing the workflow is a relevant part of a process-aware information system [BB07]. It is means of providing workflow participators with the given work items by representing a so-called *work list* [LAH08]. A work list can be regarded as an arbitrary type of workflow representation. Consequently, a *visualization model* has to be provided which determines how a structured workflow description is transformed into a visual structure. A workflow visualization component is to be regarded as central spot which gives information about all process-related data and implements the user access point to the workflow [BRB05].

Different viewpoints can be taken into account. The first viewpoint relates to the graphical representation of a predefined workflow instance in general, including work items and other relevant elements associated to annotation processes. Second, as pointed out in Section 6.1, annotators need to obtain information about the workflow state, especially in consideration of distributed workloads in collaborative use cases. Third, specific interaction facilities enable users to gain direct influence on the workflow execution, which has to be regarded as a fundamental requirement in the context of workflows with dynamic characteristics. In the following, *general requirements* for the visualization of annotation workflows will be illustrated (I.). Based on these requirements, aspects of *workflow representation* (II.), *monitoring* (III.), and *interactive interference* (IV.) are elucidated in the succeeding parts.

I. General Requirements for Workflow Visualization

Mental Mapping Support. Basically, all relevant information about a workflow need to be represented by a respective visualization component. Among these information are the tasks, connecting transitions, and specific events such as user or system-related decisions referring to the task execution sequence. In addition to that, different *Workflow Patterns* have been described in Section 2.2.5, which specify the modular operational elements of a workflow and define the process behavior at different parts of it. In this context, a central goal is to provide human participants with a general overview of the tasks to be accomplished as well as the expected sequential order. That is, users must be enabled to gain a appropriate idea or cognitive representation of the given process. Rinderle et al. and Yung et al. regard such kind of workflow representations as *mental maps* [RBRB06, YLS$^+$04].

Personalization. Especially in the scope of (asynchronous) collaborative annotation scenarios, the processes may be long-running, include a large number of activities, and involve several user groups and roles [BRB07]. In this case, users have different tasks and respective knowledge about the process [BRB05]. Furthermore, with respect to workload distribution, not only different tasks but also different contents can be assigned to single users, groups, or roles. To give a simple example, users which take the role of an *administrator* or *task leader* exclusively will need to obtain view on configuration-related tasks (cf. Section 5.2.2). As a consequence, a uniform workflow visualization for all participants can not cover all expectations and individual requirements [BRB05]. Hence, the workflow must be presented in various ways, including a personalized visualization with an appropriate level of granularity [BB07]. In this sense, views of the process must be created which only represent tasks participants are involved in, and which also show not completed parts of the workflow [BRB05]. In doing so, the specific needs of different user groups can be fulfilled.

Representation Reduction. Reporting on *process-aware systems*, Bobrik et al. pointed out that workflows are often presented to the user the same way as it has been specified by the workflow modeler [BRB07]. As a consequence, information is represented which is not relevant for the actual end user. Examples for irrelevant workflow elements can be graphical representatives for different workflow patters (cf. Section 2.2.5), such as AND-splits, OR-splits, XOR-splits, AND-joins, OR-joins, or XOR-joins (see Figure 2.16 in Section 2.2.3). Such elements refer to specific points within a workflow and indicate for its further course, at which respective decisions are mostly made by the system. Consequently, a key requirement for the visualization of the annotation workflow at run-time is a reduction of the process representation complexity, excluding not relevant information from a user-specific view. Thus, the presented workflow is visualized in a form which can be regarded as suitable for persons which do not exhibit specific knowledge about the respective (graphical) language.

Interactive Interference. Besides of system-computed decisions about the further course of the annotation workflow, there are also potential points within the workflow at which users have to made decisions about the next task or executive path to be followed. This is the case when multiple alternatives are available that acquire equal value (from a operative point of view). Moreover, in order to avoid a too stringent guidance, users need to obtain sufficient degree of freedom with respect to the selection of tasks (if logically executable). Hence, facilities for interactive interference of the annotation process are required.

Workflow State Feedback. As already pointed out in Section 6.1, features have to be offered which enable monitoring of the annotation process. This especially applies to the distribution of the annotation workload. Feedback needs to be provided during a running annotation session.

To sum up, a general requirement is the suitable visualization of the predefined annotation workflow in order to provide users with a basic overview of the included tasks and procedures, and to foster the construction of an adequate mental representation of the process. For this purpose, abstraction of the graphical representation needs to be conducted in two ways. First, personalization has to be provided with respect to different roles, tasks, and contents user can obtain in the scope of collaborative scenarios. Second, workflows generally include information which is not relevant for the end user and needs to be filtered out. Furthermore, annotators must be provided with sufficient degree of freedom with respect to the selection of tasks. Finally, feedback has to be given about the current process state, concerning any modifications made during process execution.

According to the listed requirements, different aspects of interactive workflow visualization are addressed in the following sections with respect to the *Graphical Representation of predefined Workflow Specifications*, *Visual Workflow State Monitoring*, and *Interactive Workflow Interference*.

II. Graphical Representation of predefined Workflow Specifications

As stated above, the visualization of the annotation process aims at providing a general overview of the tasks and sequences included in a specific workflow: here, contained elements are to be represented in adequate graphical manner, whereas the term adequacy refers to potential to assist human participators at constructing a correct mental representation [RBRB06, YLS+04]. In the following, existing workflow visualization approaches will be compared. Subsequently, the proposed visualization model as well as its included elements will be described.

Existing Workflow Visualization Approaches

In order to specify a visualization model, existing approaches for a graphical representation of processes have been examined. Among these are *Flow Charts*, *UML Activity Graphs*, and *Process Graphs*.

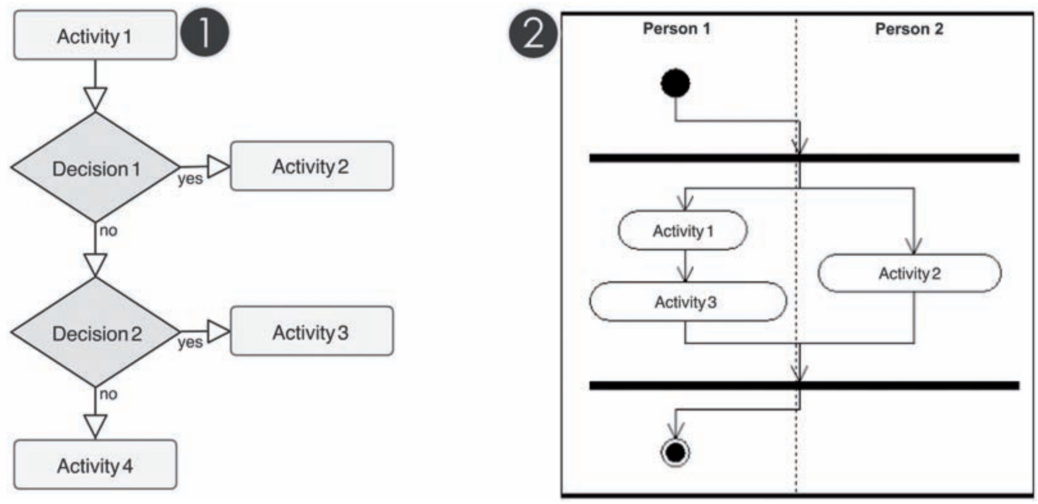


Figure 6.20: Schematic illustration of two different processes represented by a *Flowchart* (1) and an *UML Activity Graph* (2).

Flow Charts. By means of *Flow Charts*, dynamic processes can be described, such as business processes or algorithms [Far70, ST02]. As illustrated in Figure 6.20 (1), events or activities are represented as different kinds of rectangles. These may also contain different types of handled data. Additionally, rectangles are connected by arrows which represent procedural the flow and its directions. In general, flow charts are applied in different areas such as process analysis, design, or management [Far70].

UML Activity Graphs. An *UML Activity Graph* is a specific diagram type provided by the *UML 2 specification* in order to describe workflows, business process, and similar procedures. In general, the behavior of a specific process is determined by specifying sequences and conditions associated to activities. As showed in Figure 6.20 (2), an UML Graph is formed by activities are represented by nodes which are linked by edges, defining the overall flow. Activity nodes can be divided into three categories [Boc03]: *Action nodes* describe occurring events and handle received control and data values. *Control nodes* transfer control and data items through the entire graph, comprising decision-related constructs which realize the selection of alternative flow paths. *Object nodes* cache data items which wait for being passed to a successive activity, depending on specific conditions.

Process Graphs. The most applied approach for representing process models are *Process Graphs* [PW08]. Here, activities are represented by nodes which are connected by directed edges that define the specific sequential order [PW08, ST02]. Furthermore, nodes can be clustered into groups of activities with similar properties. These groups may indicate for program modules, system components, or geographic regions [ST02]. Figure 6.21 illustrates the graphical user interface of the Proviado system [BB07], which visualizes workflows by means of a specific process graph implementation.

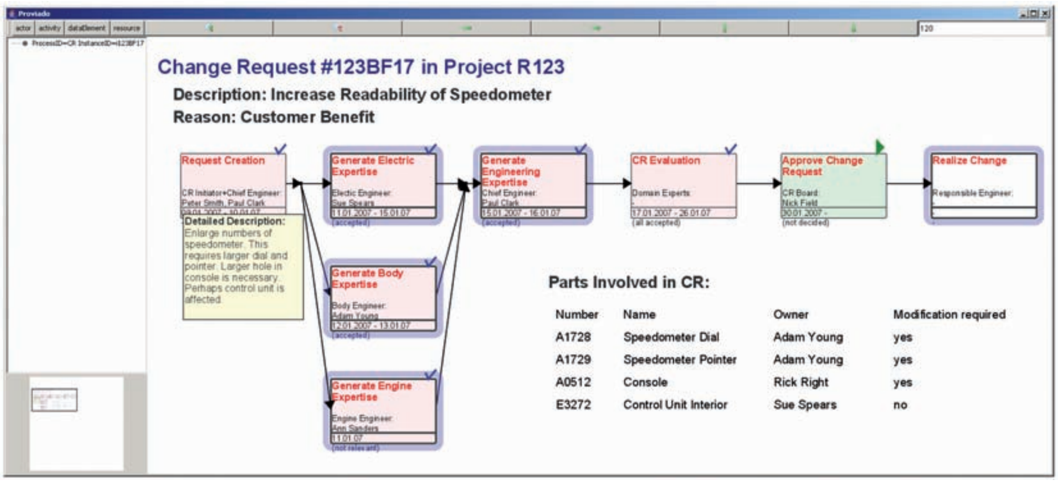


Figure 6.21: Process Graph Visualization offered by the *Proviado* System.

General commonalities can be identified at contrasting these approaches with each other. First, graphical representatives are provided for specific workflow events. These include process start and end, tasks or activities, and specific control events at which decisions are made on the further proceeding of a running workflow. Second, the transitions between events are represented, mostly by means of directed edges.

Graph-based Visualization of Multimedia Annotation Workflows

On the basis of the considerations described in the previous paragraphs, and not at least due to their level of distribution, a *Graph-based Visualization* is proposed for multimedia annotation processes. Here, this thesis refrains from treating specific approaches and algorithms for process graph construction. Respective topics are addressed for instance by Bobrik and Rinderle et al. [BB07, BRB07, RBRB06].

In Figure 6.22, the schematic structure of the proposed annotation workflow visualization is pictured. At the initial construction of the workflow graph, especially the consideration of account-based information is required in order to obtain a personalized view on the process. Consequently, it must be checked which tasks and/or media documents are probably assigned to the respective user. This can be a result of previous workload distribution on the one hand. On the other hand, this information can be derived from specific roles or groups associated with the user account. The storage and management of such kind of workflow-related information at run-time as well as on server side is addressed in part *Overall Workload Progress Monitoring* below. After conducting respective checks, the tasks, transitions, and further essential elements of the workflow to execute are set. On this basis, the graph-visualization can be constructed. Here, special focus is put on transition entities, which are typically represented as directed relations in the scope of common process definition approaches. Thus, a transition always contains a source and a destination activity, which comprise tasks and other control elements as well. In the following part, the single elements contained by a workflow graph will be elucidated.

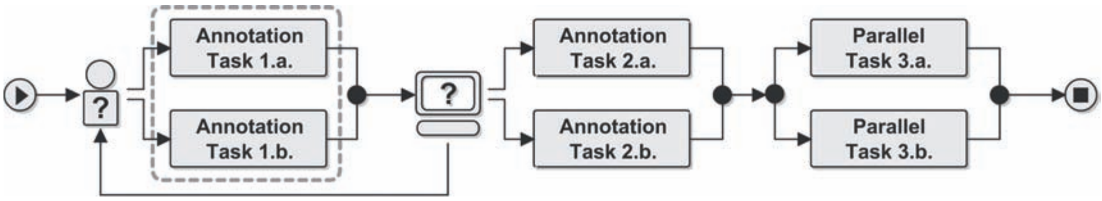


Figure 6.22: Structure and Elements of the Workflow Visualization Graph.

Abstraction of Basic Graphical Workflow Elements

Basically, the following elements included in an annotation workflow have to be considered: (i) *workflow start* and *stop* events, (ii) several kinds of *annotation tasks* which are associated with respective *services* and *users*, (iii) *control events* which may temporally interrupt the process in the case of alternative workflow threads, and may either be performed by users or the system, (iv) with respect to the latter aspect, also alternative threads must be considered which need to be executed in *parallel* (i.e., no predecision is required), and (v) *junction* of different incoming workflow threads, which may involve certain *preconditions* (e.g., all sub-processes need to be completed).

Especially the last three items are to be associated with specific *Workflow Patterns*. In this context, among the most common patterns are *Sequence*, *Parallel Split*, *Synchronization*, *Exclusive Choice*, and *Merging*. Examples for the appearance of specific patterns in annotations processes have been described in Section 2.2.5. Since workflow patterns can be regarded as descriptions of process behaviors, they are typically defined by logical constructs, at which the basic types are *AND, OR,* and *XOR splits* and *joins* (cf. Section 2.2.3).

From this the conclusion can be drawn that the annotation workflow includes various forms of information, from which several are to be regarded as not relevant for the effective end user, especially assuming that users do not prove detailed knowledge about the underlying language or coding system in the majority of cases. Consequently, the entire set of workflow-related information has to be abstracted and subsumed to few graphical representatives. For this purpose, basic elements are introduced which are integrated in the workflow graph as iconized nodes.

In this scope, especially different forms of decision-related control nodes as well as points of thread merging are represented by common graphical objects. In the following paragraphs, the incorporated iconized basic elements are elucidated, based on numbered items included in Figure 6.23. In the following, the single graphical elements are elucidated.

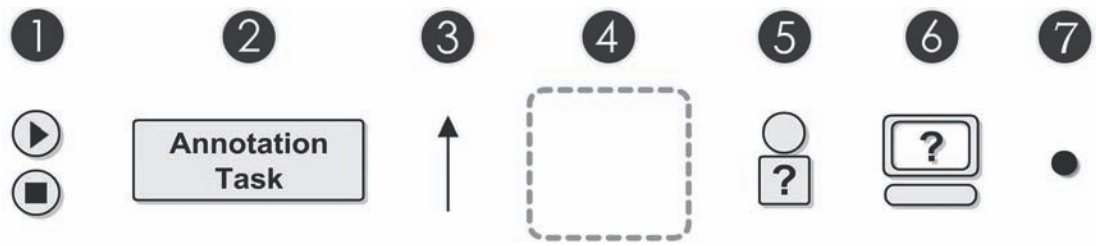


Figure 6.23: Basic Graphical Representatives for Elements of the Annotation Workflow.

(1) Process Start and End. A concrete workflow instance may intend an explicit triggering and termination of the workflow by means of a respective user interaction. For this purpose, specific start and end nodes are incorporated. This applies to graphical symbols usually offered by graphical process description notations such as BPMN or YAWL (see Section 2.2.3).

The adoption of an explicit initial interaction is founded by the general goal of the workflow visualization component to provide an overall overview. On the one hand, users which are contrasted with a new workflow need to obtain knowledge about the tasks and procedures to be forthcoming. On the other hand, with respect to asynchronous collaborative scenarios, the visualization can provide relevant information about modifications made by co-annotators during absence. This aspect will be addressed in Section *III. Workflow State Monitoring* below.

End events typically signalize the termination of a workflow-run and/or at program exit. Mostly, such kind of events are preceded by an explicit inquiry or confirmation request. If a workflow termination is signalized, the workflow visualization has to be updated or resetted, waiting for the next start command triggered by the user.

(2) Annotation Task. Specific elements are applied in order to represent annotation tasks. These items can be interpreted as entities of a process definition which are declared as *Activity* (cf. Section 2.2.1). Within a process definition, such items include a Activity Name attribute, which needs to be delivered to the graphical representative.

Potentially, tasks can be connected to specific preconditions, at which they may obtain an inactive status if execution is (temporally) not legal. For instance, a task might be only applicable to one specific media format due to its connected service. An inactive state can be symbolized by graying out or transparency, at which the specific information is not presented by default, but can be provided in the scope of element description (see the following part *Workflow Element Description*). An approach for the determination of task authorization is described in Section *IV. Interactive Workflow Interference*.

(3) Transition. Transitions between successive tasks or activities are represented by directed edges or arrows. This is also the case for recursive procedures in which bidirectional relationships are constituted. In doing so, source and destination activities are constantly defined. This principle correlates with the typical structure of process definition languages.

(4) Activity Blocks. Workflow Management allows the consideration and definition of Activity Blocks. These are to be regarded as set of activities which share common properties, so that the system shows a certain behavior with respect to the block in total [WFM99]. In the context of multimedia annotation, activity blocks can pool similar tasks and activities which belong to a common sub-process of annotation. For instance, a block might contain different tasks that provide segmentation facilities for various media formats.

Decisions. Control events can generally be regarded as decisions regarding the further workflow procedure. Here, two different types of decision events can be distinguished which refer to the determination of the workflow continuation: user-specific decisions and system-specific decisions which are computed in the background. Thus, two different symbols are adopted. As illustrated in Figure 6.23 (5) and (6), representative metaphors are proposed. According to Baecker et al. [BGBG95], metaphors contribute to the understanding of a new functional domain by employing comparable codes of already known domains.

(5) User-specific Decision. User-specific decisions are required when alternative paths or sub-processes can be followed which, due to their equality in the scope of the entire process context, can not be previously assigned to firmly defined conditions. For instance, the annotation workflow may begin with the selection of the media to be processed, at which annotators may choose between annotating new "raw" data or already edited media for inspection purposes. Accordingly, such events always indicate for alternative sub-processes or tasks which are not assigned to certain conditions. Different strategies can be pursued in order to realize user decisions. A first variant is a simple workflow interruption, at which the user is may choose one of the successive threads.

In addition to that, a second variant can be the application of specific dialogs which explicitly provide the alternative options.

(6) Automatic Continuation (System-specific Decision). System-driven decisions are always tied to specific predefined conditions or control routines, which can include exclusion, waiting times, or synchronized simultaneous execution. On behalf of process modeling, such conditions are defined by the indication of specific modules, methods, code parts, or interfaces. From the perspective of the end user, procedure computations are performed in the background on the functional concept without requirement for user interactions. Thus, the different forms of automatic continuation can be expressed by one common graphical representative.

(7) Split and Join Points. By means of the proposed symbol, generally all forms of events can be symbolized, which include the division of a single thread or the junction of multiple threads, not determined by specific conditions. Consequently, it is to indicate for a seamless continuation of the process without interruptions or system-sided background computations.

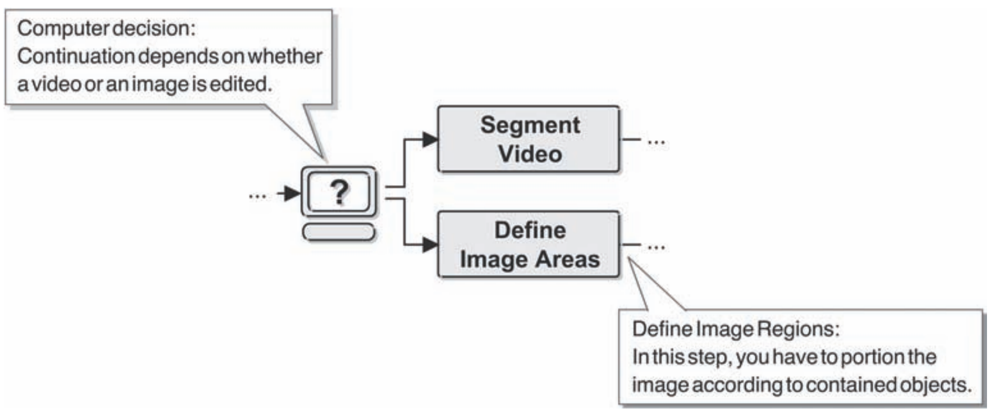


Figure 6.24: Possible variant of Workflow Element Description.

Workflow Element Description

In order to foster user orientation, overview gathering, and understanding, a description of respective basic elements is required. Among others, this can be realized by applying textual descriptions, such as offered by so-called *tooltips* (see Figure 6.24). In this manner, more detailed statements can be included, which refer to annotation activities or specific control events. In the latter case, this has to be regarded as a useful means of displaying information about the conditions or procedures computed by the system. In this context, annotation tasks can be tied to preconditions which determine whether the activity may be performed concerning the recent workflow state. Such descriptions can be placed within a respective position in the workflow specification file and need to be read and interpreted by the process visualization component.

III. Visual Workflow State Monitoring

The workflow logic permanently undergoes changes in the course of workflow execution proceeding, thus a new status is acquired after each activity [YLS$^+$04]. In other words, a certain amount of tasks and/or contents is accomplished by degrees. These changes of the workflow status need to managed by the process engine, but also have to be presented to the human participators in order to provide visual workflow monitoring. Accordingly, the progress is to be visualized with respect to an assigned or selected workload. Here, monitoring can be considered from two different viewpoints. First of all, there is an *overall workload*, which refers to the entire workload to be processed. Second, on the level of single workflow runs or sessions, a certain set of tasks has to be accomplished, e.g., for one media object passing the process from the start to the end point. Hence, also a *run-level workload* refers to the tasks that need to be edited for recently processed media documents. On the basis of this distinction, the Interactive Workflow Visualization View (cf. Section 6.4.1) is subdivided into two sub-views: *Overall Workload Progress* and *Run-level Workload Progress*.

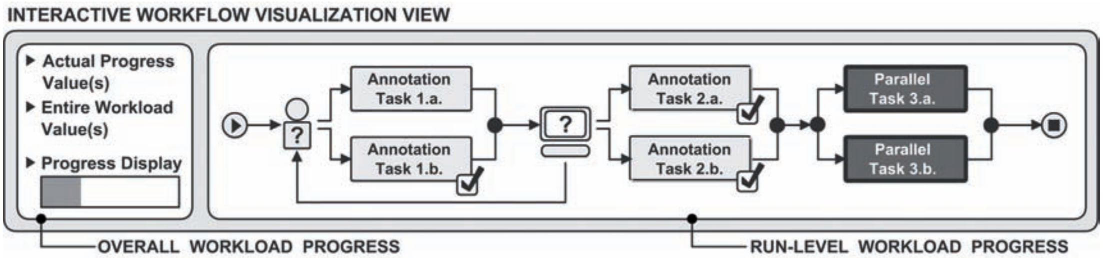


Figure 6.25: Visualization of the Process State on Overall and Run level.

Run-level Workload Progress Monitoring

As illustrated in Figure 6.25, the visualization of the workload progress on run-level is realized within the process graph. The graphical signalization of workflow updates comprises two aspects. First, once a service passes on a “completion” message to the Process Engine component, the respective graphical representative of the task associated with this service obtains a specific mark, for instance, by applying chekmark iconization. Second, the next task(s) to be accomplished, i.e., the new *recent* task(s), are highlighted. In doing so, the user obtains information about the accomplished tasks referring to the actually treated media document or set of media documents, as well as the current task which can be perceived in companion with respectively invoked and displayed service(s) (see Section 6.4.3). According to these aspects, a set of annotation tasks to accomplish is to be regarded as *task load*.

Overall Workload Progress Monitoring

A representation of the overall workload regards all tasks that have to be performed by a specific user, which result from administrative activitiesm, task distributions, or role/group assignment. Additionally, a further factor is also the number of contents which have to be annotated. These contents can either be media documents or media document subsets. This set may be a result from distribution processes or explicit selection of media entities for a forthcoming annotation session. Due to the distinction between media documents and media document subsets, this part of the entire workload is generally to be regarded as *content load*.

The entire workload may be composed of the task load and content load. Here, it is not a mere summarization, but must be explicitly computed as intersection of both sets. The reason is that tasks are connected with services which do not support all media formats comprised in the set of media entities in some cases. For instance, let it be supposed that a certain project intends the annotation of videos, images, and audio files, and one of the tasks within the workflow is "extract chords". Obviously, the service connected to this task will not support incoming video or image files. As a result, the effective number of tasks is determined by the number of *legal tasks* for each media entity registered in the content load.

As showed in Figure 6.25 above, the entire workload progress can be either represented in textual manner, for example by displaying the total number of media contents, the number of completed contents, etc., but also by means of specific graphical elements such as a progress bar. Here, the actual progress value can be calculated from the maximal extent of the respective graphical element, divided by the entire workload, that is, the total number of legal tasks. Thus one extent unit is achieved. A multiplication by the number of accomplished tasks results in the final workflow state value. Hence, the following computation has to be realized:

$$e_{act} = \frac{e_{max}}{w_{total}}|A|$$

where e_{act} is the *actual extension* (size, length, width, range, etc.) and e_{max} the *maximal extension* of the display, $A = \{a_1, a_2, ..., a_n\}$ is the set of all accomplished tasks a, and w_{total} is the *total workload*, that is, the number of all tasks which can be applied to the assigned or selected media contents. According to this, the entire workload is determined by:

$$w_{total} = \sum_{i=1}^{m} w_i$$

where w_i is *local workload* for one media content m_i. Algorithm 1 elucidates the determination of the total workload in a pseudocode view. Here, the applicability checks can be performed on the basis of service descriptions, such as described in

Section 6.2. In this context, a sophisticated implementation of service descriptions can also be means of a assigment of services at run-time.

Algorithm 1 Calculate total workload w_{total}

input: set of assigned or selected media contents m
input: set of assigned tasks t
output: total workload value w_{total}
for each media content m_i **do**
  for each task t **do**
    if t applicable to m_i **then**
      increment local workload w_i
    end if
  end for
  add local workload w_i to total workload w_{total}
end for

In addition to the points concerned up to now, a further relevant aspect is the management and storage of the workflow state in its different elucidated facets, which is to be ensured locally on the running application, as well as centralized on behalf of the server application. The latter point especially applies to long time processes, maybe in combination with asynchronous collaboration. In this case, the annotation process might be interrupted at a certain point and continued some time later. Moreover, the workflow state might have been modified by other users. Consequently, the overall workflow progress (including also the run-level state) is to be managed in centralized manner by the service application. Here, for each media content of the associated content load the local task workload $W_i = \{\{l, c, u\}, \{l, c, u\}, ..., \{l, c, u\}\}$ is saved as set of triplets that each represent one task. Here, l and c are boolean values determining if the task is legal for the specific content and if the task has been completed, and u records the user which edited the task for this content (see Table 6.3). In doing so, also user-related data can be displayed upon the workflow graph, regarding changes conducted by co-annotators within a shared annotation workflow. In the context of change tracking for collaborative annotation scenarios, a detailed approach is addressed in Hofmann et al. [HBF10] with respect to the specific challenges of video-based media. The basic principles of change tracking in collaborative workspaces are described in [TG06]. Moreover, Papadopoulou [Pap09] presented a framework that supports change awareness for collaborative text authoring work settings.

IV. Interactive Workflow Interference

Interactive access of human participators on the given workflow instance is a further relevant topic that has to be concerned in the scope of the visualization of multimedia annotation workflows [BRB05]. Here, regarding the objective to achieve a reduction or

	task$_1$	**task**$_2$	...	**task**$_n$
content$_1$	$\{l, c, u\}_{1,1}$	$\{l, c, u\}_{1,2}$	...	$\{l, c, u\}_{1,n}$
content$_2$	$\{l, c, u\}_{2,1}$	$\{l, c, u\}_{2,2}$	...	$\{l, c, u\}_{2,n}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
content$_i$	$\{l, c, u\}_{i,1}$	$\{l, c, u\}_{i,2}$	...	$\{l, c, u\}_{i,n}$

Table 6.3: A basic Scheme for Workflow State Storage.

granularization of the level of user guidance, selective activities are focused by which users can initiate the execution of specific tasks in self-motivated manner. Two main arguments can be quoted in this context: First, several studies on user guidance in comparable application scenarios revealed that a strong guidance (if well-implemented) can be helpful for novice users, but more experienced persons rather prefer the admission of more degrees of freedom [SBCO01, ZZZ07]. In addition to that, the conducted studies on annotation practices (see Chapter 4) showed that jumps, returns, and iterations must be considered, which arise from explorative activities such as the comparison of other results. Thus, user might need to improve or correct already generated data. Consequently, as long as no automatic mechanisms of data validation are incorporated, it must be assumed that jumps and iterations need to be performed in user-centered manual manner.

Two different levels of user guidance are defined: *hybrid mode*, and *silent mode*. In the hybrid mode, the workflow support is activated, that is, workflow execution is realized by the Process Engine, selective interaction is allowed by means of the Process Visualization Component and, additionally, a "typical" selection of tools is facilitated by means of classical interaction elements. Referring to the latter aspect, as will be described below, the synchronization of the workflow visualization and potentially integrated menus, toolboxes etc. must be ensured. In the silent mode, the workflow visualization as well as service supply are deactivated, that is, no explicit user guidance is provided. Here, the Workflow Handler Component only listens to specific activities in order to update the workflow state in the event that the workflow support is reactivated by the user. Furthermore, the workflow state is still required for tool selection item activation on respective UI components. In addition to that, no service invocation requests are sent to the Component Broker instance.

With respect to user-centered workflow control, a fundamental requirement is the consideration of tasks which may not be selected in the scope of a specific workflow state. This results from the specific characteristics of workloads defined in the scope of annotation workflows which comprise several media formats. As explained in the previous part, this refers to whether task can be regarded as legal for editing a specific media format, depending on the associated annotation service. Accordingly, certain tasks need to obtain a *inactive* or *not executable* state. Thus, the state of a single task or activity depends on the recent workflow state, so that it must be explicitly computed at each transition between two activities. This state is to be concerned by incorporated graphical task items. For this purpose, the required information can be gathered from the updated workload state model, such as schematized in Table 6.3 before. As pictured in Figure 6.26, different strategies can be applied. Among these are the increase of transparency values, gray out, or resizing.

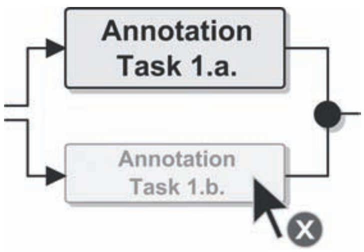


Figure 6.26: Signalling of temporally not legal tasks depending on incorporated media formats or task preconditions.

Another factor affecting the selective access on graphical task elements can also be specific task preconditions, which potentially can be defined in the scope of workflow modeling. To give an example, a certain task B succeeds the parallel execution of several tasks $A_{1,\ldots,n}$. A possible precondition is that all incoming tasks must have been accomplished, before the execution of task B is legal. Hence, a succeeding task needs to be hold at an inactive state until all services connected to the previous tasks report back.

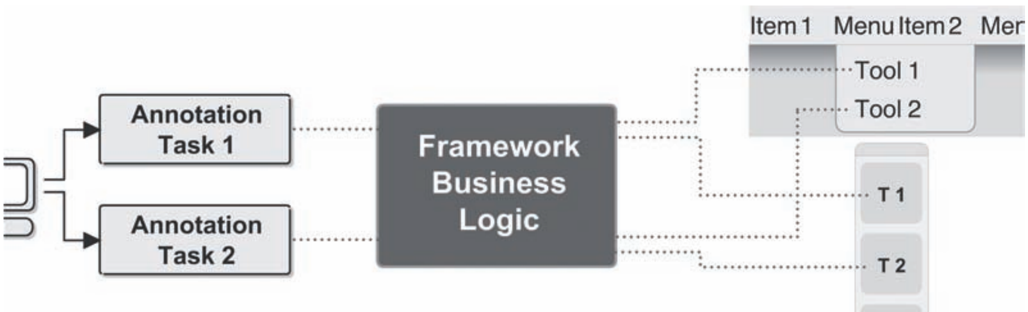


Figure 6.27: Synchronized treatment of task items within Workflow Visualization and Traditional Tool Selection components.

Equally respecting the differentiation between multiple level of user guidance, interactive UI elements have to be taken into account which have been regarded as means of *traditional tool selection* in Section 6.4.1. As a result, the framework must ensure the synchronization between such elements with an integrated Process Visualization Component, at which synonymic entities from both sets of service-related task elements and menu or toolbar items are brought into compliance (see Figure 6.27). Additionally, it becomes clear that task activation as described above is to be mapped on classical tool selection elements in forms of *tool activation*.

6.4.3 User Guidance by Workflow-Driven Service Supply

Besides of Interactive Workflow Visualization, the proposed concept for Process-driven User Assistance also intends an explicit supply of services depending on the recent annotation task to perform. In this context, the internal framework procedures of service invocation based on workflow definitions have been described in Section 6.3.4. An explanation of the invocation and placement of such kind of transient services on the graphical user interface has been given in Section 6.4.1. Nevertheless, this section summarizes the basic idea of workflow-driven service supply and clarifies the relationships between the introduced workflow visualization instance and invoked services.

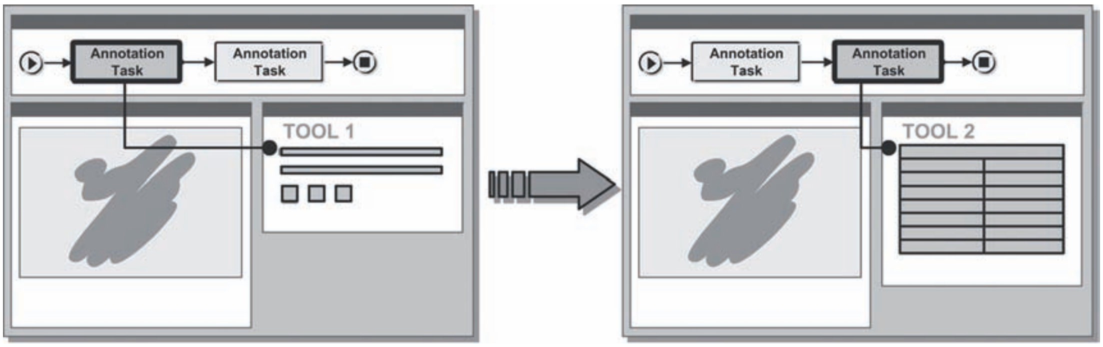


Figure 6.28: Outline of the basic idea of Workflow-Driven Service Supply.

As described in the previous sections, the graph-based visualization provides user guidance with respect to the process sequence. At the same time, it includes facilities for user-specific interactive control of the workflow, so that different degrees of guidance are realized. Thus, the established concepts must be regarded as *assistance* rather than *guidance*. User assistance not only relies on presenting or visualizing workflow-related information, parts of it effectively intend to constitute explicit guidance with respect to the acquired annotation services. Thus, as illustrated in Figure 6.28, the basic idea is to invoke and close services (that is, the respective UI component) once the next task has been selected or automatically determined. At the same time, the respective task is highlighted in the context of the entire workflow.

The main goal of this idea is the implementation of seamless procedure of multimedia annotation, regarding the operation of a respective UI. Here, the explicit supply of annotation services is combined with a representation of the entire workflow and its current state on run-time, so that a visual and thus cognitive connection is established between relative entities of both areas. In this context, incorporated methods for user guidance can lead to an alleviation of learning the system's operation and an improvement the actual operation of the user interface as well [BP05]. According to this, an early study on user guidance revealed that, if guidance is well designed and implemented, it can accelerate processes through faster task performance, reduce errors, and enable novice users to perform tasks of information handling they usually would not have been able to perform at this state of experience [Mag83]. The reason is that the cognitive overhead is reduced at determining the next tools or service to apply. Second, additional orientation is provided with respect to the workflow procedure, that is, the set of single steps which are connected with tasks and respective UI components. Hence, a greater user satisfaction can be achieved [Mag83].

6.4.4 Summary

In this section, a visual-interactive concept for *Process-driven User Assistance* has been presented. The main goal is to support users during the execution of (collaborative) multimedia annotation processes, at which process-relevant information is provided. Among this information are (i) all annotation tasks which have to be accomplished, (ii) their sequential arrangement, (iii) the current state, i.e., the recent task to perform, (iv) the available tools or services which comply to the recent task, (v) which tasks have already been accomplished, and (vi) which tasks still lay ahead.

In this context, a basic visual design is presented in Section 6.4.1, at which the framework UI is subdivided into four functional areas. Accordingly, the proposed *4 View Model* includes a *Traditional Tool Selection View*, in which classical interaction elements for the selection of system functionalities, such as menu bars or tool boxes, are included. An *Interactive Workflow Visualization View* comprises components that realize a graphical representation of annotation workflow instances, additionally providing workflow monitoring facilities as well as interactive human access. Moreover, incorporated annotation services are placed either upon a *Permanent Services View* or a *Transient Services View*. Here, a distinction is drawn between services which are required during an entire annotation run and services that are invoked and closed according to the recent annotation task.

Section 6.4.2 describes a concept for *Interactive Workflow Visualization*. In particular, a graph-based visualization model is proposed, based on the consideration of existing workflow visualization approaches. In this scope, basic graph elements have been specified, considering the fundamental elements included in an annotation workflow. For this purpose, different workflow events and execution patters have been analyzed and subsequently subsumed to graphical representatives which are to be regarded as the most essential for human participators. In doing so, the understanding and traceability of the given workflow is supported for end users which do not hold detailed knowledge about specific (visual) workflow modeling codes and languages.

Furthermore, it has been pointed out that a workflow permanently changes during its execution, since tasks and media documents are continuously edited. According to this, an approach has been developed which guarantees *Visual Workflow State Monitoring*. Here, it is distinguished between *Run-level* and *Overall Workflow Monitoring*. The first item relates to the processing of tasks during a single annotation run. In contrast, the overall workload refers to the entirety of tasks and/or media contents which are to be edited during the whole annotation process (e.g., a project). With respect to the overall workflow load, a solution approach has been described that meets specific challenges of multimedia annotation workflows, particularly considering that tasks need to be deactivated in some cases, depending on the associated annotation service.

The third aspect assigned to workflow visualization is a identified requirement for an interactive intervention in the running workflow procedure. The two main reasons are that potential acts of re-editing must be considered on the one hand, and on the other hand a too high level of user guidance potentially leads to negative effects. Hence, different levels of user guidance have been specified, emphasizing the cooperation and synchronization of traditional tool selection components and workflow visualization elements.

Besides of the visualization of annotation tasks and progress, Section 6.4.3 extends the concept for user assistance by a method for explicit *Service Supply* during workflow runtime. Here, the basic idea is to invoke or close annotation services (i.e., respective UI components), depending on whether they can be applied to the current annotation task. In doing so, a seamless procedure that task execution is achieved, aiming at reducing users' cognitive load regarding the operation of a given user interface.

6.5 Conclusions

In this chapter, a conceptual framework for the design of process-based multimedia annotation systems has been established. Initially, several requirements have been derived which exclusively refer to the specific challenges regarding multimedia annotation workflows and collaborative work settings. In the following sections, the developed solution concept has been elucidated. Here, a distinction between three different constituent concept part has been drawn: a data modeling and management concept, an architectural concept, and a visual-interactive concept. In doing so, different aspects and layers regarding a realizable overall system have been covered. These areas of conceptual contributions will be summarized in the following, checking the fulfillment of requirements defined in Section 6.1. A tabulated illustration of the results is showed in Table 6.4 at the end of this section.

Data Modeling and Management Concept

In the context of a data modeling and management concept, a *Formal Specification of Annotation Processes* has been constructed in Section 6.2, which has to be regarded as a workflow scheme adapted to scenarios of (collaborative) multimedia annotation. Here, basic elements included in annotation processes have been defined, considering specific properties and relations to other elements. In doing so, relevant classes of process-related information entities have been declared, which need to be processed, visualized, and made accessible for users by a process-based annotation system.

According to the drafted requirement for *Process Definition* (R1) regarding a use case-specific predefinition of individual annotation workflows, a framework of guidelines has been established which can be employed in the scope of workflow modeling processes, also defining general rules for *Workload Distribution* (R8).

Architectural Concept

On the definition of a *Reference Architecture Model*, initially the basic structure of the environment has been elucidated. Through the arrangement by a Client-Server Model, a central server application has been incorporated which adopts essential functions for *Data Consistency* (R6), *User Management* (R11), *Data Exchange and Sharing* (R12), and *Correct Data Handling* (R13) in the scope of the entire environment. Moreover, the realization of Fat-Client applications places essential process-related functionalities on accessing clients.

An Adapted Model-View-Controller and Mediator Model includes different local controllers and one central global controller instance on behalf of the system business logic. The global controller represents the process engine and provides features for workflow interpretation and execution, as well as integration, coordination, and synchronization of incorporated annotation services and further components. Consequently, detected requirements of *Workflow Control* (R2), *Flexible Service Integration* (R4), *Data Consistency* within client applications (R6), and *Workflow-driven Service Invocation* have been met.

With regard to *Sub-Process Enclosure* (R3), multiple local controllers serve as interfaces between (annotation) services and the entire framework as elements within a component-based environment. Here, specific functions are provided which refer to the special properties of annotation sub-processes presented in Section 5.4. Furthermore, local controllers are connected to modules incorporated in the global controller instance, which hold read and write access on single parts of an applied annotation data model. Hence, *Coherent Data Junction and Management* (R7) have been ensured.

Visual-interactive Concept

Regarded as the main contribution of this thesis with respect to problem areas of multimedia annotation elucidated in Section 1.1, a concept for *Process-driven User Assistance* has been proposed in order to support human workflow participators during the conduction of the annotation process on behalf of a graphical user interface. First of all, as part of the Basic Visual Design, a 4 View Model has been presented which subdivides the UI into four functional areas. In particular, an *Interactive Involvement of Users* (R10) has been considered by providing space for traditional tool selection components (menus, toolbars, etc.). Additionally, a spatial UI area has been defined for the placement of transient annotation services, which applies to *Workflow-driven Service Invocation* (R5).

Moreover, the concept includes a graph-based visualization of the workflow, considering comprised elements and respective sequential relations. In addition to that, the workflow state and progress is represented in graphical manner on different levels of a given workload. Hence, basic requirements of *Process Visualization and Monitoring* (R9) have been fulfilled. Additional *Interactive Involvement of Users* (R10) is realized by providing facilities for a manual selection of annotation tasks, if they are legal with regard to the current workflow state. In this context, different levels of workflow support are offered in order to avoid a too strong degree of user guidance. Finally, *Workflow-driven Service Invocation* (R5) is provided on behalf of the graphical user interface by *supplying* (displaying and hiding) UI representatives of annotation services depending on the recent task to perform.

Requirements		Conceptual Contributions	Sections
R1 R8	Process Definition Workload Distribution	***Formal Specification of Annotation Processes***	6.2
		Reference Architecture Model	
R6 R11 R12 R13	Data Consistency User Management Data Exchange and Sharing Correct Data Handling	Client-Server Model	6.3.1
R2 R4 R5 R6	Workflow Control Flexible Service Integration Workflow-driven Service Invocation Data Consistency (Client)	Adapted MVC and Mediator	6.3.2
R3 R7	Sub-Process Enclosure Coherent Data Junction and Management	Component-based Environment	6.3.3
R7	Coherent Data Junction and Management	Architectural Structure for Data Junction	6.3.5
		Process-driven User Assistance	
R5 R10	Workflow-driven Service Invocation Interactive Involvement of Users	Basic Visual Design	6.4.1
R9 R10	Process Visualization and Monitoring Interactive Involvement of Users	Interactive Workflow Visualization	6.4.2
R5	Workflow-driven Service Invocation	User Guidance by WF-driven Service Supply	6.4.3

Table 6.4: Fulfillment of defined Requirements by the partial Solution Concepts developed in the scope of this thesis.

Chapter 7

SemAnnot: Semantic Annotation of Multimedia Documents

The three constituent solution concepts established in this thesis form the basis of *SemAnnot*, a framework and toolset for a process-driven semantic annotation of multimedia documents. In this context, different annotation tasks can be defined and executed in automatic manner, supporting various steps between the incorporation of raw media, preprocessing and low-level descriptive annotation, and high-level semantic annotation. SemAnnot is a partial framework of *SemaVis*[1], a modular and adaptive framework for visualization and processing of semantic information [Sem].

This chapter is organized as follows. In the first section, technologies applied for system implementation will be described. Next, the general structure of the SemAnnot client application will be elucidated in Section 7.2. Section 7.3 especially focuses on different implemented annotation components, and Section 7.4 will show how two different annotation workflows have been realized based on the integrated components. In Section 7.5, results of an independent usability test on SemAnnot will be presented. Finally, a summary of presented implementation aspects will be given in Section 7.6.

7.1 Applied Technologies

The description of technologies applied at system development has to be subdivided into three parts, according to the design and implementation of both a client and a server application, as well as the individual specification of annotation workflows by means of process definition techniques.

[1]SemaVis, including SemAnnot, is developed at the Fraunhofer IDG Darmstadt in the context of the *THESEUS Core Technology Cluster* (CTC). THESEUS is a research program initiated by the Federal Ministry of Economy and Technology (BMWi) to develop a new internet-based infrastructure in order to better use and utilize the knowledge available on the internet [The].

7.1.1 Client Application

The SemAnnot client has been developed as web application and can be executed within standard web browsers. The implementation has been realized by means of *Adobe Flex* [Flaa], a platform independent open source framework for the development of rich internet applications (RIA), based on the *Adobe Flash* platform. Thus, the realization of a Fat-Client application is enabled, allowing a placement of multiple annotation and workflow-related functions on the client side. The SemAnnot web application can be executed with any web browser including an installed *Adobe Flash Player* runtime environment which, according to an examination on distribution conducted in March 2010, is applied at 99 % of all computing devices and operating systems with internet access [Flab].

Formally produced and distributed by Macromedia, Adobe Flash is an authoring platform for the development of interactive web presentations. Besides of UI design and the processing of vector graphics, it enables a script-based integration of interactive elements and the communication with external applications and distributed servers. Here, *ActionScript 3* is applied, which has evolved from script-based to an imperative programming language which implements a wide range of principles from object oriented design since version 2. In addition to ActionScript classes, Flex provides the incorporation of *MXML* components. MXML is an xml-based declarative language which, similar to HTML, allows the description of visual and not visual components, defining the structure of the application or component UI. MXML files are transformed into ActionScript source files, which are both compiled to Flash files (.swf) by the Flex Compiler. The graphical user interface of the SemAnnot client application is illustrated in Figure 7.1.

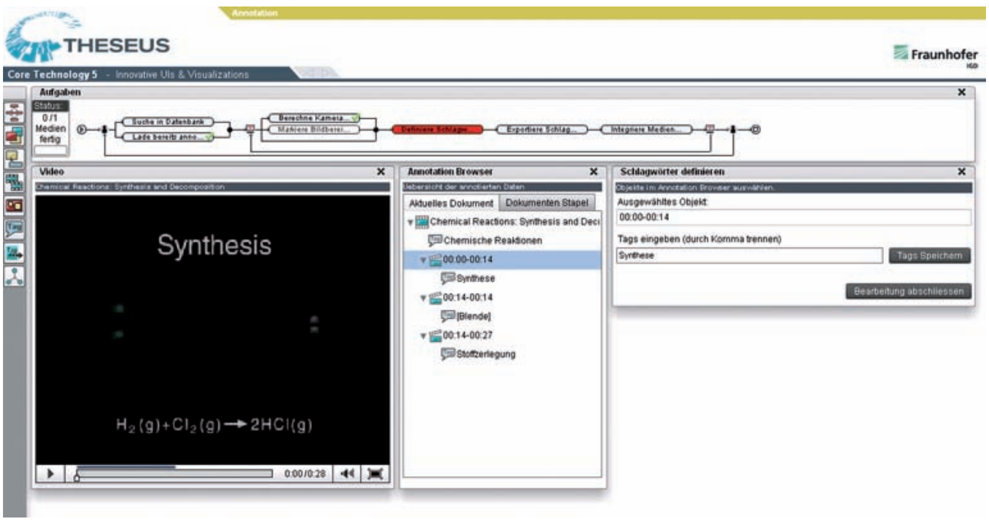


Figure 7.1: Graphical User Interface of the SemAnnot web application.

7.1.2 Server Application

The implemented server application is responsible for centralized storage and management of relevant information, such as media files (physical as well as references), annotation data, users, and workflow definitions. Access to these data units is realized by the provision of specific Java Web Services. According to the W3C, the provision of web services supports an integration and cooperation of different applications or components, which can be located and run on varying distributed platforms within a service-oriented architecture (SOA) [W3Cb]. A web service is regarded as a software application or component, which is identified by a URI and provides xml-based interfaces for definition, description, and localization purposes. Here, the communication with further software agents is fostered by means of xml-based messages which are transfered accompanied by specific internet protocols.

According to the latter point, the access to server-centered web service is implemented by the employment of the *Simple Object Access Protocol* (SOAP), a network protocol for data exchange and remote procedure calls between different peer systems [W3Ca]. SOAP relies both on the standards XML for data representation and internet protocols of the transport layer of the TCP/IP reference model to perform message transmission. A SOAP message is structured according a head-body-model and contains an Envelope, a Header, and a Body area. The SOAP-Envelope field serves as general container, while the Header part includes metadata such as routing or encoding information. The Body area provides the effective payload data. The different data access methods provided by the server can be addressed by client applications according to the described structure. For this purpose, the message body includes the name of a service or method, and optionally may consist of additional parameters, for instance, explicit numerical values for new anchors. The header part incorporates metainformation about objects the invoked service is to be applied on. In general, *get*, *add*, *modify*, and *delete* methods may be called. Listing 7.1 shows an exemplary call of media documents already set up in the server context.

The implemented Multimedia Annotation Server is based on *Jetty Web Server*, a Java framework which provides an HTTP server, an HTTP client (not used in this case), as well as a Servlet and JSP container for web service modularization. Both static and dynamic content can be served either from standalone or embedded components [Jet]. Here, the Apache Tomcat JSP compiler *Jasper* is applied. All elements are open source and distributed in the scope of the Apache license. Data storage is implemented by the employment of an *db4o* database developed by *db4objects Inc* [Db4]. Db4o is an open source object-oriented database is provided by Java and MS .NET platforms, and relies on the data model of a respective application language, so that object-relational mismatches are omitted. Among the main advantages of db4o databases are a fast persistence of complex objects, efficient query languages, low administration costs, and diverse options for specific configuration. Finally, the binding between server application and database is realized by *dom4j* [DOM], a Java enabled open source library especially designed for working with XML, XPath and XSLT.

```
<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
  encoding/">

<SOAP-ENV:Header>
  <documentId xsi:type="xsd:int">19</documentId>
</SOAP-ENV:Header>

<SOAP-ENV:Body>
   <hvi:fatclient.GetDocument>
   </hvi:fatclient.GetDocument>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 7.1: SOAP encoded call of the GetDocument service.

7.1.3 Individual Definition of Annotation Workflows

An essential goal of this thesis is the support of different use cases by permitting an individual predefinition of annotation workflows. In the specific case of semantic multimedia annotation, also multiple use cases and respective workflows can be identified. As will be described in Section 7.4 below, two different workflows are realized in the context of SemAnnot.

For workflow specification purposes, the two modeling and definition standards *Business Process Modeling Notation* (BPMN) and *XML Process Definition Language* (XPDL) are applied. Section 2.2.3 illustrated both standards in detail, including an explanation of their specific cooperation and mapping. Hence, their integration is explained briefly in the following, focusing on their specific role within the realized implementation.

Workflow modeling is performed by means of the graphical process definition standard BPMN. This notation provides graphical elements in order to represent entities within business processes (in this case, annotation processes). Respective workspaces include a flowchart representation. At the present moment, respective authoring process are realized by means of the *Together XPDL Workflow Editor* application [Tog], a Java-based editor which creates XPDL specifications. In Figure 2.15 (Section 2.2.3), an excerption of general mappings from BPMN to XPDL was illustrated. XPDL is applied in order to interpret and execute the predefined workflow instance by the system. It is an XML-based format for the description of business processes. Here, workflow elements are organized in linear structure, defining activities, transitions, conditions, services, and users (also groups and roles), implementing their association by means of unique IDs. This structure is read and interpreted by the SemAnnot Workflow Handler component, after previous transformation by an XPDL-specific parser module.

7.2 General Client Structure

The SemAnnot client is basically realized by the exploitation of so-called *Flex MXML Components*. According to the concept of software components explained in Section 6.3.3, MXML components are implemented as integrated and enclosed units which provide specific functionalities and an own graphical user interface. In addition to that, system components which are to be associated with the system business logic are implemented based on ActionScript classes. As shown in Figure 7.2, among the main packages of the client architecture are (annotation) components, the process engine, the applied data model classes, specific session data, and a connection package containing protocol-specific classes for server communication.

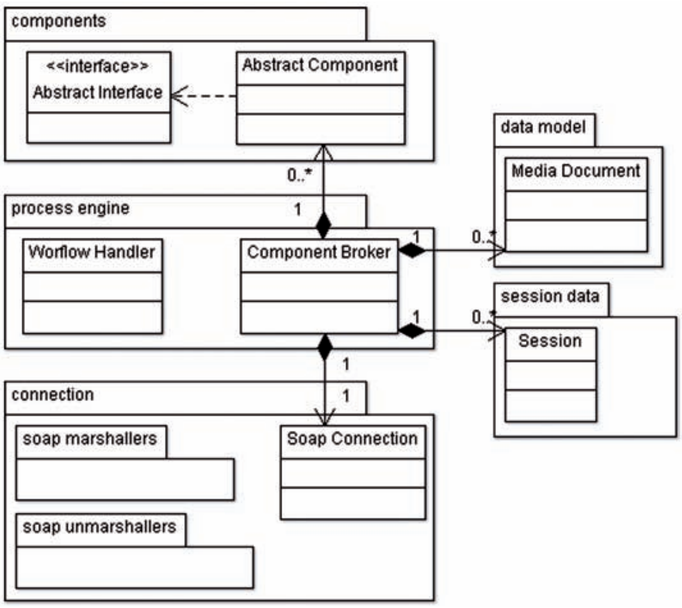


Figure 7.2: General Architecture of the SemAnnot client.

(Annotation) Components. As will be elucidated in the next section, annotation components establish the integration and access to the annotation services applied in a specific workflow. Here, a superior component Abstract Component defines essential common properties. The basic functionalities are determined by specific interfaces. This especially concerns the different activities and data forms of annotation, which are involved during process execution.

Process Engine. The process engine package comprises the two classes Workflow Handler and Component Broker, forming the central control unit with in the MVC and Mediator model. As described in Section 6.3, the Workflow Handler is primarily responsible for the interpretation and execution of a predefined workflow definition, obtaining access to the integrated workflow visualization component. On the other hand, the Component Broker manages and controls all components to be invoked, and is additionally responsible for synchronization and event forwarding, as well as coherent junction of heterogeneous annotation-related data. Thus, it has exclusive access to the applied data model.

Data Model. In order to reduce client-server communication and respective not necessary waiting times, the client application manages a local data model, which maps the central data set stored on behalf of the server machine. Here, the superordinate instances are one or more objects from the Media Document class, which serves as container for all other classes defining media files, annotations, and relations between them. A general description of the applied data model has been given in Section 6.3.5.

Session Data. The session data package pools procedures for the runtime management of data related to the current client-server session. Among these data are login specific data and a session id. Furthermore, workflow-related data is processed, such as a currently specified workload and, respectively, a local copy of the recent workflow state.

Connection. The Soap Connection class included by the connection package implements the communication between client and server application, including specific access methods which are based on the applied SOAP protocol. For this purpose, SOAP-specific classes of marshallers and unmarshallers are integrated, so that a transformation between Flex objects and SOAP messages is ensured.

7.3 Employed Annotation Components

SemAnnot has not only to be regarded as pure framework, it also provides a set of annotation-related tools (as enclosed components) in different predefined workflow scenarios. These components are assigned to the (annotation) components package introduced above. As illustrated in Figure 7.3, all components are generalizations of the superior component Abstract Component. Here, general common functions are define which determine the invocation (open), concealing (close), and the general initialization (init) of an annotation component. Concerning the latter point, the initialization implies the registration on the Component Broker, at which a reference to the broker instance is generated. Here, the visual UI part of a component is defined by an implementation as MXML component. In addition to that, the local controller building block which serves as interface between the annotation component and the broker instance is implemented by a separate ActionScript class. Furthermore, interfaces are defined which refer to different forms of features and data types which need to be supported. Thus, interfaces are based on the several process-related components introduced in Section 6.3.4.

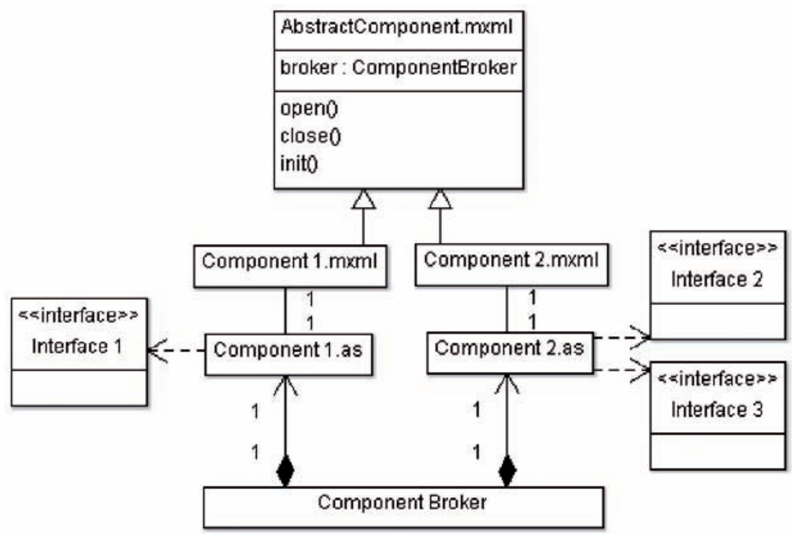


Figure 7.3: General structure of integrated Annotation Components.

The set of incorporated components can be illustrated based on the structure of the main UI panel, which has been developed according to the 4 View Model proposed in Section 6.4.1. Here, a subdivision is conducted between the (i) Traditional Tool Selection View, (ii) an Interactive Workflow Visualization View, (iii) a Permanent Services View, and (iv) a Transient Services View.

Both the Traditional Tool Selection View and the Interactive Workflow Visualization View each contain one component, *Toolbox* and *Workflow Graph*. In the Permanent Services View, particularly all media display components are integrated. Here, the supported media formats are processed by the components *Video Player*, *Image Display*, *Audio Player*, *Flash Player*, and *URL Display*. In addition to that, media display components are always accompanied by the *Annotation Browser* panel, which represents all data generated during the current or a previous annotation process execution. In addition to that, components for media selection are integrated, either for search and load media files from external resources (*Media Browser*) or files already annotated and registered within the annotation framework (*Annotated Media*). Finally, the Transient Services View obtains the different incorporated tools for annotation which are invoked and hidden during workflow execution. Among these tools are *Cuts Computation*, *Image Area Marker*, *Tagging*, *Tag Export*, and *Semantic Visualization and Editing*.

Toolbox Component

The *Toolbox Component* provides access to all tools (services) which can be manually selected by the user and are integrated in the context of the loaded workflow definition. In this context, all available services are checked at initialization time. with respect to specific workflow rules which determine the admissibility of selection according to an actual workflow state, tool representatives are set into an disabled state when required. A tool selection event is directly sent to the Component Broker, which subsequently performs correct service invocation.

Workflow Graph Component

The *Workflow Graph Component* implements the concepts established for Interactive Workflow Visualization as described in Section 6.4.2. Figure 7.4 illustrates the workflow graph, which primarily represents all tasks to be performed, their sequential order, as well as specific control points in which human or automatic decisions on the further workflow procedure are to be conducted. Furthermore, event or element descriptions which have been specified within the workflow definition file can be read by means of tooltips displayed after a mouse roll-over interaction of a duration of more than 500 ms. The current workflow state on *run-level* is represented first by highlighting the current task in red, signing already accomplished tasks with a green checkmark, and bleaching not legal tasks by gray out and transparency. The *overall workflow state* is represented in the left margin, at which the total number of tasks accomplished for the total number of assigned media documents is showed by means of a textual representation as well as a graphical progress bar.

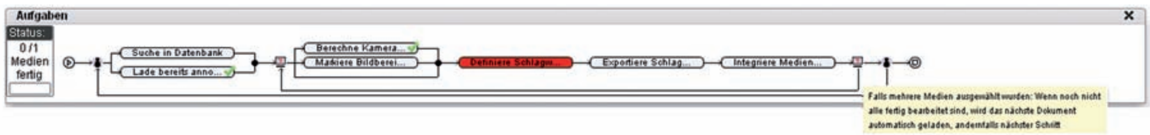


Figure 7.4: User Interface of the Workflow Graph component.

Media Display Components

Media Display components are responsible for the display or playback, control, and navigation of media files which are annotated during workflow execution. Up to now, supported media formats are video (.flv), image (.jpg,.gif.,.png), audio (.mp3), flash presentations (.swf), and url-referenceable content (.doc,.pdf,.ppt,etc.). For this purpose, Figure 7.5 illustrates the integrated components which include *Video Player* (1), *Image Display* (2), *Audio Player* (3), *Flash Player* (4), and *URL Display* (5). While the Flash Player is limited to just a pure representation of delivered media objects, the remaining components provide additional media-specific functions and interactions elements. The Video Player offers video controls for play and pause, a slide bar to perform forward and rewind actions, time representation, a volume control, and a full-screen feature. Image Display is enabled to highlight image marks which have been defined as spatial anchors by means of the Image Marker component explained below. Similar to Video, the Audio Player provides a play and pause button, as well as a control slider bar. Furthermore, URL Display provides a button that enables users to open the respective document in a separate web browser window. In doing so, document formats which are not explicitly supported by SemAnnot can be integrated and annotated on entire document level.

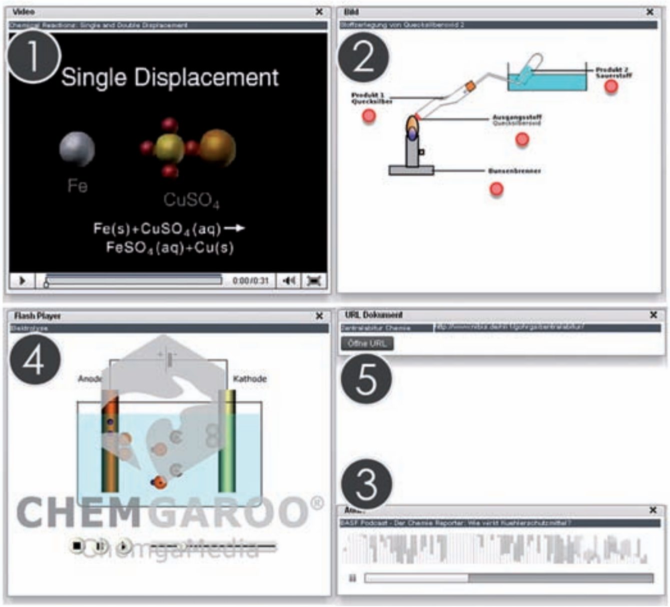


Figure 7.5: Incorporated Media Display components: Video Player (1), Image Display (2), Audio Player (3), Flash Player (4), and URL Display (5).

Annotation Browser Component

The *Annotation Browser* component basically visualizes the entire annotation structure for media documents processed within the process. Here, two different panels are provided. The first one shows the annotation structure of a currently active media document by means of a tree visualization. Figure 7.6 exemplifies the display for a loaded video document. The applied tree pictures the entire video, defined video segments, as well as descriptive keywords attached to both types of entities. Different icons symbolize the various potential forms of media and annotation entities. This panel has to be regarded as central point during the execution of the annotation process. It serves as monitoring component on the one hand, since any new generated information is represented. On the other side, it provides facilities to interact with other components, so that the processing-flow for an permanently enriched information space is supported.

The second panel gives an overview of the current content load, that is, all media documents which are processed in the workflow. Here, different icons are applied which represent several media formats. Once a media document is selected, the component switches to the primary panel, showing the specific annotation structure.

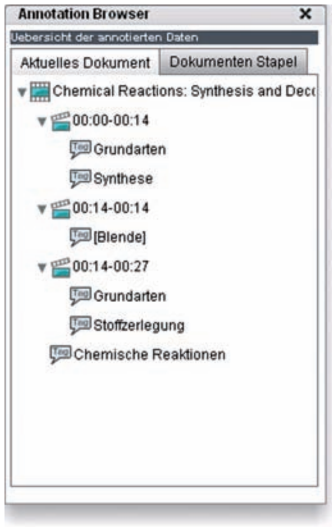


Figure 7.6: The Annotation Browser component visualized the entire annotation structure of media documents.

Media Selection Components

In order to select media documents to be processed, the two components *Media Browser* and *Annotated Media* have been developed. The Media Browser component is employed in order to select media documents made available by different multimedia resources or databases. That is, the component is responsible for gathering of "raw data". For this purpose, it expects an XML-structured listing, from which also potentially existing metadata can be extracted, such as title and description. The Media Browser tool supports multiple selection of media files, so that the specification of a *content load* (cf. Section 6.4.2) is enabled.

The Annotated Media component facilitates the selection of media documents which have already been edited with SemAnnot (but also other clients cooperating with the server) in previous sessions. Among others, this tool can be applied in order to perform successive modifications or corrections on the given data set. As shown in Figure 7.7, the right half of the panel provides a preview area for selected media objects, at which several media display or playback elements are offered for the different types of supported media formats. The same applies to the Media Browser tool.

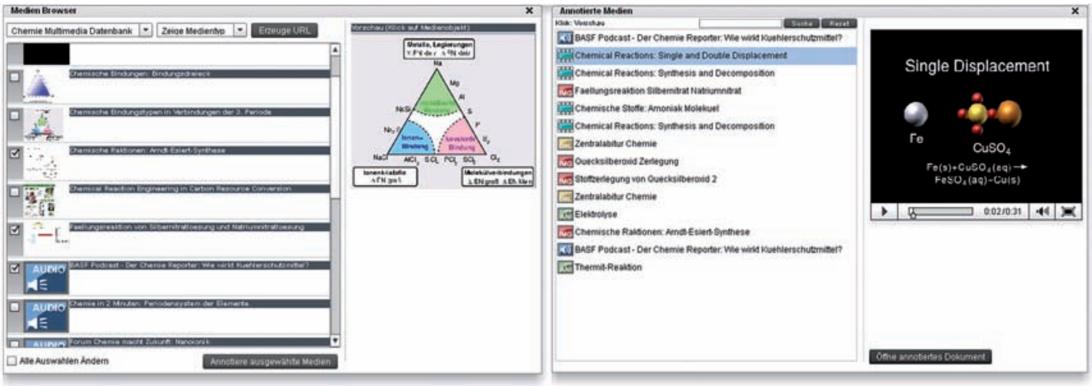


Figure 7.7: Media Selection by means of the Media Browser and Annotated Media components.

Media Segmentation Components

Segmentation of media is applied in order to enable a separate annotation of partial areas within documents. Here, SemAnnot supports video shot detection and image area marking. For video shot detection purposes, the *Cuts Computation* component has been integrated (see Figure 7.8(1)). This tool is connected to an external shot detection service (2) which expects video URLs and returns time interval values. Additionally, the Cuts Computation component provides interaction elements to configure the sensibility value for shot computation, and a timline-based display of generated shot intervals. SemAnnot also allows marking of point areas within images. In this scope, the *Image Area Marker* component (3) stores point coordinates in cooperation with the *Image Display* tool (4), which enables users to manually mark points by means of mouse clicking. In order to define an entire document as annotation validity area, SemAnnot by default creates a *Generic Anchor* (cf. Section 6.3.5) for each loaded document at its initialization time.

Tagging Component

As already explained in Section 2.1.2, tagging is a useful means of classifying and organizing digital objects. The incorporated *Tagging* component permits the keyword-based classification of objects previously selected in the Annotation Browser. Here, multiple tags, separated by commas, can be defined for one single object.

Tag Export Component

By means of the *Tag Export* tool, a cooperation with the *Sophie* framework has been established. Generally, this framework is used for a semi-automatic analysis and modeling of semantic structures [BFar]. The connection between the two frameworks thus

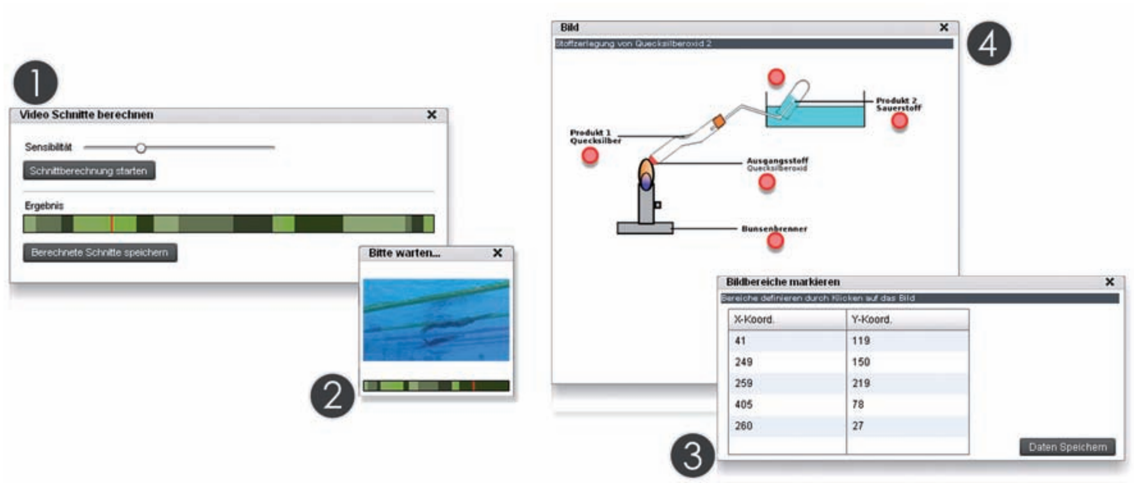


Figure 7.8: Media Segmentation components for Shot Detection (1),(2) and Image Area Marking (3),(4).

supports a semi-automatic generation of semantics, based on tags previously defined within the internal SemAnnot workflow. Here, the Tag Export component exports selected media and/or media subsets (by indication of an URL) as well as respectively attached tags in forms of a text file. This file is subsequently read by Sophie, at which an analysis of the delivered tags is exploited to generate semantic concepts and relations associated to the given media objects.

Semantic Visualization and Association Component

The *Semantic Visualization and Association* component is exploited in order to visualize a given semantic structure on the one hand, and on the other hand to associate media documents or document subsets as instance of a given semantic concept. As illustrated in Figure 7.9, two different visualizations are applied: *SeMap* (top) and *SemaGraph* (bottom). SeMap provides an hierarchical representation of concepts and relations, and is primarily used for association in the SemAnnot context. SemaGraph additionally displays instances, so that it can be used for monitoring purposes. Both visualizations are provided by the superordinate *SemaVis* framework and embedded in the component panel at runtime[2]. The semantic association is realized through Drag&Drop from the Annotation Browser tool onto the respective concept within the visualization.

[2] It is to be noted that the SemaVis framework actually provides significantly more visualizations and features than those applied in the SemAnnot context. More information can be gathered at [NBB$^+$09].

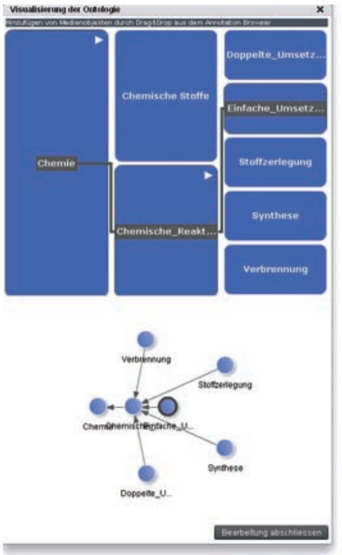


Figure 7.9: The Semantic Visualization and Association Component embeds Visualizations provided by the superordinate *SemaVis* framework.

7.4 Implemented Annotation Workflows

Two different concrete workflows have been implemented for the conduction of digital multimedia (semantic) annotation. Both workflows realize a semantic annotation of multimedia documents, at which various steps are traversed starting with the selection of raw media, and continuing with the attachment of low-level descriptive data and finally higher-level semantic information. In the following, the two workflows will be described.

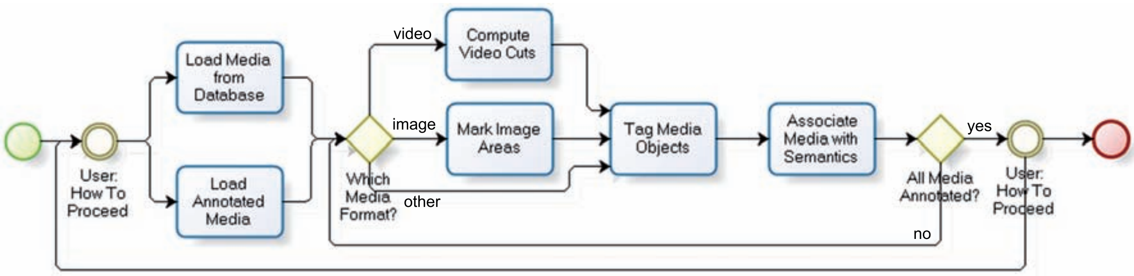


Figure 7.10: Annotation Workflow 1: from Raw Data to Semantic Infrastructures.

Workflow 1. As illustrated in Figure 7.10, the first workflow starts with a user-centered decision, at which the human participator needs to select whether unmachined files or already annotated documents are to be loaded. Here, multiple files of different media formats can be specified by means of the *Media Browser* or the *Annotated Data* tools. The decision request is conducted by means of a specific dialog (see Figure 7.11). The second cluster of annotation tasks refers to the segmentation of processed media files. Here, before proceeding with the specific task, the system first checks the media format of the currently passed document. In case of videos, the workflows proceeds with "Compute Video Cuts" and invokes the *Cuts Computation* component. For images, the user has to mark specific image areas using the integrated *Image Area Marker* component. For all other supported media file formats, no segmentation tool is provided. In this case, the workflow continues with the third task "Tag Media Objects", which is to be performed with the *Tagging* component. Subsequently, in the scope of task four "Associate Media with Semantics", media documents and subsets are associated with semantic concepts provided by the *SemaVis* visualizations embedded in the *Semantic Visualization and Association* component, at which the multimedia contents are defined as instances of respective concepts. In the next step, the application checks whether there are still not completely processed media documents from the previously selected set of files. As long as the entire set is not accomplished, the system loads the next media document and returns to the second task cluster "Media Segmentation". When all files have been accomplished, the user is asked via dialog whether the workflow is to be restarted for a new set of media files, or if it is to be terminated.

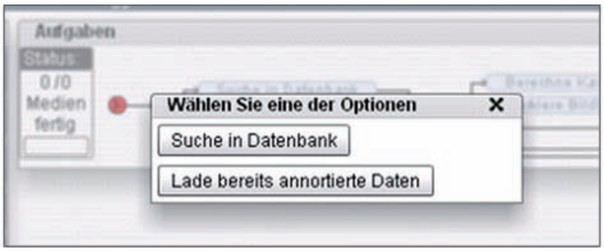


Figure 7.11: Dialog-based request of user-centered workflow proceeding decisions.

Workflow 2. The second workflow generally replaces a manual association of media objects with semantic concepts, as supported by workflow 1, by a semi-automatic generation of a semantic structure based on specified tags. For this purpose, as described in section 7.3 above, the *Tag Export* component is employed which cooperates with the *Sophie* framework. Figure 7.12 shows the second implemented annotation workflow instance. It reveals that the first three tasks or task groups, "Media Selection", "Media Segmentation", and "Tag Media Objects", are identical to the first workflow. After accomplishing the tagging task, the system conducts a loop as long as not all previously selected media documents have been tagged. Otherwise, the workflow proceeds to the next task "Generate Semantic Structure", at which a text file containing media contents and attached tags is delivered to *Sophie*. After the tag-based semi-automatic definition of a semantic structure (in other words, after semi-automatic semantic annotation), the user may decide whether the generated semantic is to be returned to the SemAnnot environment, for instance, in order to view and potentially perform manual modifications by means of the *Semantic Visualization and Association* tool. Alternatively, the generated semantics file can be delivered directly to SemaVis, resulting in the termination of the SemAnnot workflow.

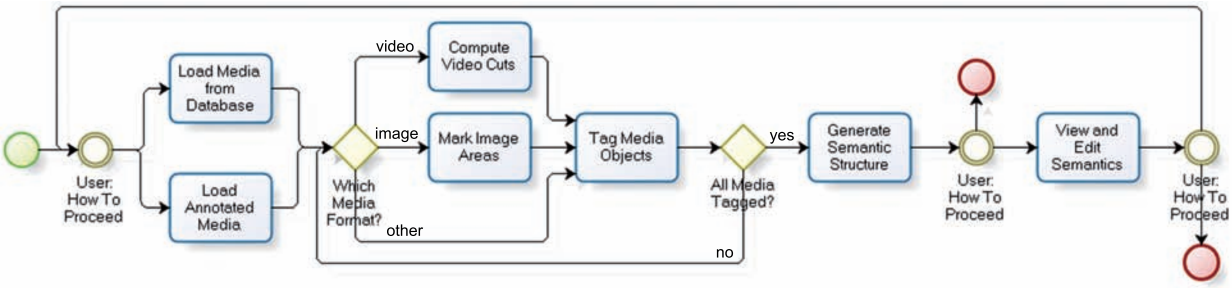


Figure 7.12: Annotation Workflow 2: Additional Semi-automatic Semantics Generation in cooperation with *Sophie*.

7.5 Usability Test

In addition to the user study performed in the scope of this thesis (see Chapter 8), an informal usability test on SemAnnot was performed at the Fraunhofer Institute for Digital Media Technology IDMT in Ilmenau, Germany, acting as project partners within the THESEUS research program. In the test, ten participants with no or only small knowledge on semantic browsing and editing where asked to perform simple annotation tasks and express their observations and feelings. In this context, workflow 1 for semantic annotation described in Section 7.4 was applied. The task was to perform an entire workflow run for one image, and successively load the annotated image at the end. With respect to task processing, the general reported outcome was that all users completed the tasks without complications, and that interface elements clearly show how interaction needs to be performed in order to process the tasks. A detailed description of the conducted usability test is provided in [KBS10].

7.6 Summary

In this chapter, SemAnnot has been presented, a framework which supports a process-driven semantic annotation of digital multimedia documents, implying the essential solution concepts for a process-based design of multimedia annotation system established in this thesis. The following implementation aspects have been especially pointed out: (i) applied technologies at system development, (ii) definition of annotation workflows by means of process definition standards and interpretation by the system, (iii) the general structure of the SemAnnot web client, (iv) implemented annotation components, and (v) two different realized workflows.

Both Section 7.1 and 7.2 have revealed how the *Reference Architecture Model* constituted in Section 6.3 was technically realized in the scope of SemAnnot. Here, Section 7.2 focused on the organizational structure of the implemented client application. According to the proposed architectural concept, a enclosed modularization was introduced which separates annotation components, a process engine comprised of a workflow handler and a component broker, as well as a communication unit which ensures information exchange between clients and server based on the SOAP protocol. Furthermore, additional packets were incorporated for the management of a local data model and session-related data, such as recent workflow states. Among others, technologies applied for the development of a Multimedia Annotation Server have been listed in Section 7.1. Here, special focus was put on the elucidation of how the server application provides read and write access on all workflow-relevant data according to the principles of web services. In addition to that, it has been described by which process definition standards (BMPN and XPDL) and editors users are enabled to specify individual annotation workflow instances. In doing so, the requirement for a support of different use cases has been met.

The description of all annotation components provided by SemAnnot (Section 7.3) illustrated that several services and tools for different forms of annotation could be integrated in defined workflows. In addition to that, it has been shown that varying component types are supported. Among these are services which provide an own UI (e.g., Semantic Visualization and Association), services without a graphical user interface (shot detection), for which SemAnnot implements a specific graphical panel, as well as stand-alone applications such as *Sophie*. This could be achieved by the realization of the structure of integrated components presented in Section 6.3.4, at which a component is subdivided into a view and a local controller part. Moreover, all introduced components have been assigned to elements of a *4 View Model* described in Section 6.4.1, which structures the basic UI into four functional areas.

Finally, two different workflows realized for semantic multimedia annotation have been described in Section 7.4. In doing so, the key requirement for the support of varying use scenarios of multimedia annotation could be complied.

The effective implementation of essential solution concepts for a process-based design of multimedia annotation systems formed the basis for a verification of the main hypotheses pursued in this thesis, regarding an improvement of the learnability and usability of respective graphical user interfaces. This verification was performed by means of a conducted User Study, which will be elucidated in the following chapter.

Chapter 8

User Study

This chapter presents an evaluation of the established concept for *Process-driven User Assistance* based on its realization within *SemAnnot*. As visual-interactive concept for the design of user interfaces provided by process-based multimedia annotation systems, process-driven assistance is to be regarded as central contribution of this thesis. Hence, the evaluation was conducted in forms of a user study, especially focusing on the comparison of two variants of SemAnnot, with and without assistance.

Single aspects to be investigated address the different expected benefits explained in Section 1.2. Correspondingly, the key hypothesis is that support of users during operation of multimedia annotation software by means of process-driven user assistance will lead to an improvement of system usage regarding the following aspects:

1. *Efficiency* with respect to the required time when executing an annotation process.
2. *Learnability* regarding the suitability of an user interface for learning its functions and interactions required at its operation.
3. *Usability*, especially concerning the system's suitability for accomplishment of the given task as well as self-descriptiveness of the UI.
4. *Process Overview*, including the user's mental representation of an annotation workflow which has to be performed.
5. *User Satisfaction*, as a general result of improvements related to the previous viewpoints.

In Section 8.1, the applied evaluation method and general study design will be described. Section 8.2 will illustrate obtained results considering the different examined sub-aspects. Summing up, conclusions will be drawn in Section 8.3.

8.1 Evaluation Method and Study Design

The conducted user study included 34 psychology students in the Bachelor programme at the Technische Universität Darmstadt, Germany. The entire group of test subjects was composed of 27 female and 7 male persons aging from 19 to 39 with an average of 23,3 years.

A 2x2 factorial design was applied with the factors *workflow support* (yes/no) and *order of condition* (workflow support first/ workflow support second), with *workflow support* as a repeated measurements factor. Here, the set of persons was divided into two groups of each 17 members, enabling to perform the study by means of the *Cross Evaluation* method. This approach is a combination of the *Within-Groups Design* and *Between-Groups Design* methods. The main characteristic is that different factors can be excluded which may lead to negative effects, for instance, learn processes in the course of the first run, which can have impact on the second run. Accordingly, the test subjects had to operate two different variants of SemAnnot in the scope of two analogue task processing runs, including the same scenario and tasks, and the consideration of homogeneous media documents and annotations. Here, the variant A *with workflow support* included explicit system operation accompanied by the workflow visualization component and workflow-driven service supply. In the second variant B *without workflow support*, the workflow visualization component was disabled and tool (and task) selection had to be performed by means of the provided toolbox. In this context, the two implemented modes *hybrid* and *silent* (cf. Section 6.4.2), which realize different levels of user guidance, could be exploited. According to cross evaluation, the first user group α started with variant A and afterwards performed the tasks of variant B, while the second group β proceeded in reversed order (see Figure 8.1).

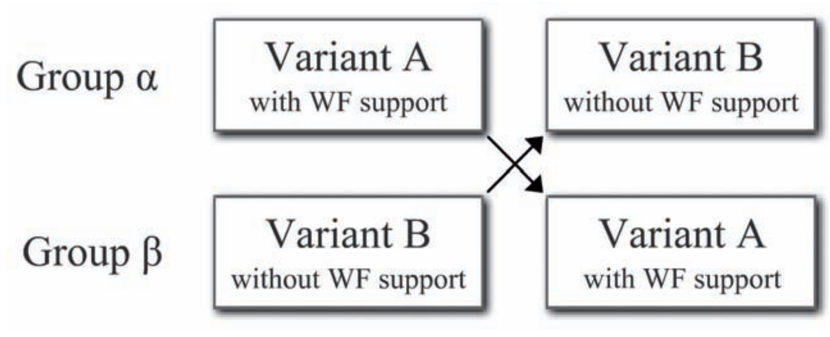


Figure 8.1: Principle of the Cross Evaluation approach.

Scenario and Tasks

The study was performed in the context of a valid scenario. The simulated case intended the test subjects to be employees of an arbitrary library which, besides of printed documents, offers an online catalogue of multimedia documents. Here, visitors can explore and retrieve these documents by means of a specific navigation panel and a search form. The general task was to integrate three videos and two images dealing with the common topics "basic types of chemical reactions" and "chemical substances", at which different forms of annotations needed to be generated in order to ensure later retrieval through the navigation and search elements. Here, similar media files were selected for both runs, including the duration of videos, the number of "annotate-able" areas (entire document, video scenes, image areas), and the thematic content with respect to the similarity of annotations to be created. The general conceptual formulation included the following subtasks:

1. *Selection of Media Documents*. The first task was to load the videos and images from a specific database. For this purpose, the *SemAnnot Media Browser* tool had to be applied.

2. *Specification of Document Areas*. The second task included the definition of document areas which were separately annotated in the following steps[1]. In the case of videos, this was achieved by means of the *Cuts Computation* (shot detection) tool. Image Areas were defined by using the *Image Area Marker* component.

3. *Classification by Keywords*. In order to simulate the specification of keywords which can be exploited by a search engine, one or more tags had to be defined for each document area by means of the *Tagging* component.

4. *Integration in the given Topic Structure*. Next, by applying the *Semantic Visualization and Association* component, all document areas were assigned to concepts (topics) provided by the *SemaVis* visualizations. In this context, the superior subject was chemistry, containing the sub-topics chemical substances and chemical reactions. Moreover, chemical reactions was subdivided into the sub-concepts (types of chemical reactions) synthesis, decomposition, single displacement, double displacement, and combustion.

5. *Additional Task*. Once steps one to four had been accomplished, an additional task was to simulate the subsequent correction of a typing error occurred during tagging of a video. For this purpose, the respective video file had to be opened within the *Annotated Media* component, proceeding with the invocation of the *Tagging* tool to perform the correction for a fixed video shot.

The described tasks were analogically performed in both variants, that is, task one to five were conducted each for three videos and two images. Obviously, this procedure

[1] As explained in Section 7.3, an entire document can be addressed by default through the automatic generation of *Generic Anchors*.

refers to Annotation Workflow 1 implemented in the SemAnnot context, as illustrated in Figure 7.10, Section 7.4. The only difference between both variants based on the fact that SemAnnot supports the selection of multiple media documents at the first workflow step (if workflow support is activated), so that all media files were selected at the same time in task 1 at variant A (i.e., task 1 had to be performed once), while each media file had to be individually opened after task 4 at variant B.

During task processing, users had two types of additives at their disposal. First, a task sheet explained the general scenario as well as the subtasks to conduct including the tools to be used for each task. Second, a tabular showing all data which needed to be generated in the course of each subtask was displayed on a second screen. That is, all segments and annotations to attach were predefined. In doing so, effects which result from different previous knowledge could be decimated. In fact, assuming that a segmentation of media objects multiplies the number of annotations attached in the successive steps, a study session took approximately one and a half hours on average, in which a test subject had to create exactly 83 annotations including segments, tags, and semantic instance declarations. The described task sheets and tables of predetermined annotations for both variants are illustrated in Appendix C.

Collection of Quantitative and Qualitative Data

First of all, time-based quantitative data was collected referring to the duration of (i) an entire annotation run in minutes, (ii) the elapsed time between the termination of tool A and invocation of tool B (i.e., the sum of all tool or task transition durations) in seconds, and (iii) the difference between both values in minutes, which can be regarded as the time when working with the single annotation components. For this purpose, the SemAnnot user interface was recorded with the screencasting software Camstasia Studio [Cam] at both annotation runs. Afterwards, the session-related video files were processed by means of *Anvil* [Kip08, Anv], a freely available video annotation and analysis application. Here, a timeline component allows the definition of layers representing different categories (e.g., *transition tool A to Tool B*, *transition tool B to tool C*, etc.), at which time intervals can be determined representing the appearance of a category within video playback time.

Additionally, three different questionnaires had to be completed. The first two questionnaires refer to the two provided variants of SemAnnot. As illustrated in Appendix C, parts 1 to 3 were identical in both variants, inquiring personal ratings with respect to (i) user satisfaction, (ii) usability, and (iii) specific aspects concerning the provision of an overview of the tasks to perform and the tools to be applied. Hence, these areas could be directly referred to each other.

For user satisfaction and usability rating purposes, incorporated questions were taken from both the *Questionnaire for User Interface Satisfaction* (QUIS) [Shn98], and the standard *ISO 9241-110:2006 Ergonomics of human-system interaction – Part 110: Di-*

alogue principles, especially concerning the issues *suitability for the task* (suitability regarding user tasks and skills), *self-descriptiveness* (indication of the next steps to perform), and *suitability for learning* (support learning of system operation) [ISO]. Furthermore, persons had to assess items regarding the general process support for both variants. The questionnaire applied for variant A (with workflow support) additionally asked for aspects exclusively referring to the provided methods for interactive workflow visualization and explicit supply of annotation services.

Finally, the third questionnaire regarded general topics, such as competencies in computer operation, previous knowledge with respect to the topics "basic types of chemical reactions" and "chemical substances", general difficulties during system operation, comparative ratings of both variants, and the explicit valuation of the realized concept for process-based multimedia annotation.

All questionnaires were composed of multiple choice questions. Moreover, a qualitative assessment was conducted by integrating additional parts of free-text estimation. The multiple choice questions permitted single selection, at which five properties were provided for subjective answering, so that test subjects obtained the possibility to place a neutral rating. Schematically, the five alternative answers covered the levels "strongly agree" and "strongly disagree". The described questionnaires are illustrated in Appendix C.

As further form of acquired quantitative data, errors or uncertainties at the selection of the next tool to invoke (or task to perform) were recorded during system operation based on check lists. Here, errors refer to the selection of the wrong component depending on the recent workflow state. Situations were rated as uncertainty, if the test subject searched the component in the toolbox by mouse scanning, passing the correct tool more than twice without selection, or, if a respective verbal statement was explicitly made.

Experiment Procedure

The sequential procedure of a study session was held in identical manner for both groups. As illustrated in Figure 8.2, the first step included a general introduction, at which fundamental aspects of multimedia annotation, as well as the main tasks and procedure of the study were explained.

After this, the test subjects received the task sheet document to read the conceptual formulation and the single subtasks. Subsequently, depending on whether the person was assigned to group α or β, the respective application variant was introduced by explaining the incorporated tools according the the respective subtask, and performing a test run exemplary for one video and one image file. In doing so, a thorough introduction was given for both variants in order to prevent potential learn effects in the first run which could have had impact on the second run within one session. In this case, such learn effects refer to the operation of the system.

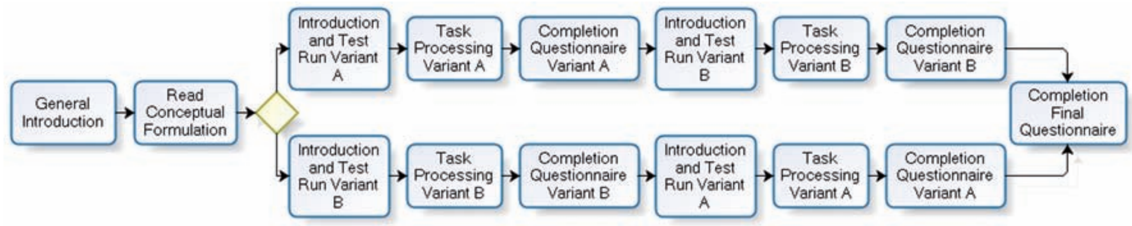


Figure 8.2: Procedure of a single run of the conducted User Study.

Next, the actual processing of annotation tasks was performed for the first application variant, at which all subtasks determined in the provided task sheet were accomplished by degrees. In the process, one exceptional case was induced. Here, one of the videos was composed of only one scene, so that the shot detection task had to be skipped, directly continuing with tagging. In doing so, the test subjects were instructed to perform a tool or task selection differing from the general task sequence formulation. This particularly aimed at showing that the provided means of workflow support also enable users to select tasks in a self-motivated manner.

Once all tasks for the first variant were accomplished, the test subjects had to fill in the respective questionnaire. Afterwards, introduction, task processing, and questionnaire completion were performed for the second application variant. In the last step of the entire study session, the final questionnaire was filled in.

Interpretation Methods

For the interpretation of time-based data for entire duration, transitions duration, and duration difference (working with the tools), means were first computed. A comparison between the two variants depending on the execution order was performed by means of a mixed design analysis of variance.

The same applies to the evaluation of usability, general process support, and user satisfaction, inquired in the common part 1 to 3 of the variant-related questionnaires. As opposed to time data, summarizing mean values were created for each scale before analyzing variances. Here, the scales were previously tested for reliability, at which sufficient Cronbach's α values $< .6$ were obtained.

Although already included in two items within questionnaire part 2 (usability), learnability was separately considered, since it is regarded as particular aspect in the scope of process support. These items include the following topics: "The user interface is easy to learn" and "The user interface does not require to memorize too many details". The items were evaluated by means of a mixed design analysis of variance.

In order to analyze the provided methods for interactive workflow visualization and service supply, errors and uncertainties at tool or task selection, as well as contrasting items in the final questionnaire, means and standard deviations were calculated.

8.2 Results and Discussion

In this section, achieved results will be presented and discussed. In the scope of the illustration of results which arise from performed variance analyses, the so-called *F-value* and *p-value* will be indicated, e.g., $F(1, 32) = 171.55$, $p < .001$. The F-test statistic compares the variance induced by the treatment to the variance induced by other factors (the error variance). It is characterized by two values (in parentheses): the degrees of freedom of the treatment variance (number of compared groups - 1), and the degrees of freedom of the error variance (number of test subjects - number of groups).

In order to identify whether the value of the F statistic is significant (groups differ significantly according to the treatment they received), the value needs to be compared to given tables of the *F-distribution*. The F-distribution depends on the given degrees of freedom. Specific values in these tables indicate specific levels of the α error. The α error occurs when a significant difference between groups is assumed, but in reality, groups do not differ. If the F-value is larger than a specific value in the F-distribution (for example at the α error level 0.05), this indicates that the treatment groups differ significantly, and that the probability that such an F-value occurs in case of a non-existent difference is 0.05. Thus, a comparison of the F-value to a given F-distribution results in p, at which $p < .05$ indicates for a significant variance and $p < .001$ indicates for a highly significant variance [Bor77].

Furthermore, it is important to mention that F-values and significance p will be indicated for each of the factors (i) workflow support (yes/no), (ii) order of condition (group α or β), and also (iii) the interaction between both factors.

In the following, only statistical values for significant effects $p < .05$ or highly significant effects $p < .001$ will be reported. According to the listing of single aspects to be investigated, results will be presented for *Efficiency*, *Learnability*, *Usability*, *Process Overview*, and *User Satisfaction*.

8.2.1 Efficiency

Efficiency refers to the duration of specific processes. In this scope, it was distinguished between (i) duration of the entire run, (ii) duration of task transitions, and (iii) the time required when working with single annotation tools.

Entire Annotation Run. For the duration of an entire run, there was a significant main effect for workflow support, $F(1, 32) = 171.55$, $p < .001$. Summarizing group α and β, subjects required less time with variant A with workflow support ($M = 12.21$, $SD = 1.78$) than with variant B without support ($M = 16.21$, $SD = 2.91$). There was also a significant interaction effect between support and execution order, $F(1, 32) = 41.48$, $p < .001$. This indicates that the order of conditions had impact on

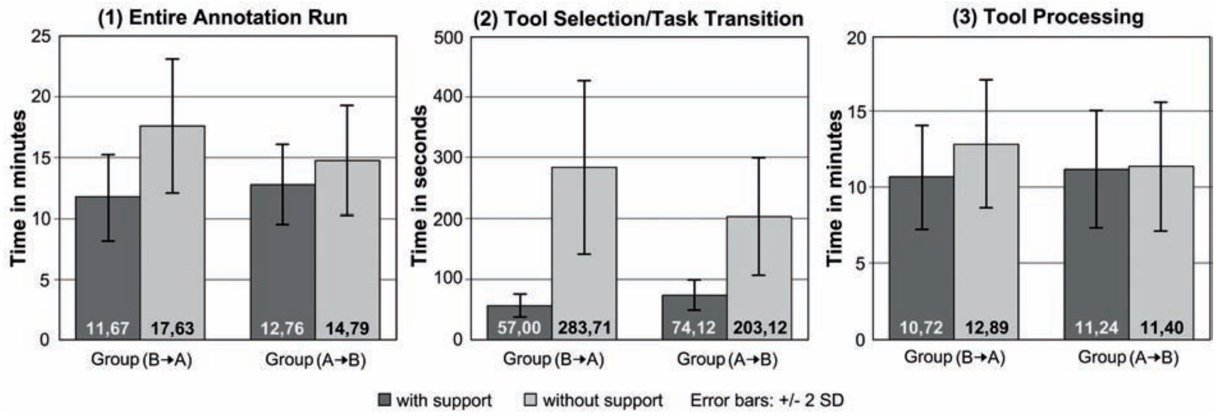


Figure 8.3: Comparison of time values for (1) an entire annotation run, (2) tool or task transitions, and (3) tool processing.

the total run duration. As showed in Figure 8.3(1), each variant was processed quicker when applied first, at which the larger difference is to be identified at variant B without support. That is, working without support led to shorter entire run durations, if the variant with workflow support was executed first.

Tool Selection / Task Transitions. As opposed to the other two cases, tool selection or task transition time was measured in seconds. Here, a significant effect emerged for workflow support, $F(1, 32) = 308.57$, $p < .001$. Working with support required less time ($M = 65.56$, $SD = 13.68$) in comparison with the variant without support ($M = 243.41$, $SD = 72.97$). The order of execution had a significant effect for support, $F(1, 32) = 758.60$, $p < .05$. As task transition times were constant for workflow support due to service supply (except for the task skip and additional task), this shows that selection of the next tool at variant B required less time, if variant A was applied previously. Figure 8.3(2) reveals that the same effects occurred for variant A with support (since also manual task selection was performed in two cases). With respect to the interaction between workflow support and condition order, there was a significant effect, $F(1, 32) = 23.28$, $p < .001$. This indicates that the condition order had impact on the effect achieved for workflow support.

Tool Processing. There was a significant effect for workflow support, $F(1, 32) = 15.49$, $p < .001$. The variant with support required less time ($M = 10.98$, $SD = 1.82$) compared without support ($M = 12.15$, $SD = 2.22$). Also a significant interaction effect was achieved between workflow support and variant order, $F(1, 32) = 11.56$, $p < .05$. This indicates that workflow support had different effects on the required time, depending on which order of conditions was used. In this context, Figure 8.3(3) reveals that working without workflow support requires less time, if one has worked with support first. This indicates that workflow support improved learning processes regarding the operation of single tools, which will be discussed in the next paragraph.

8.2.2 Learnability

Learnability was measured by a direct comparison in the scope of the final questionnaire and by two respective items in part 2 within the two variant-related questionnaires. Moreover, as mentioned above, faster processing times with respect to the duration of working with the tools indicate for improved learning effects with respect to system operation.

Explicit Decision. In answer to the question which user interface variant is better to learn, 29 respondents opted for workflow support (85.3%), 5 persons prefered the variant without support (14.7%).

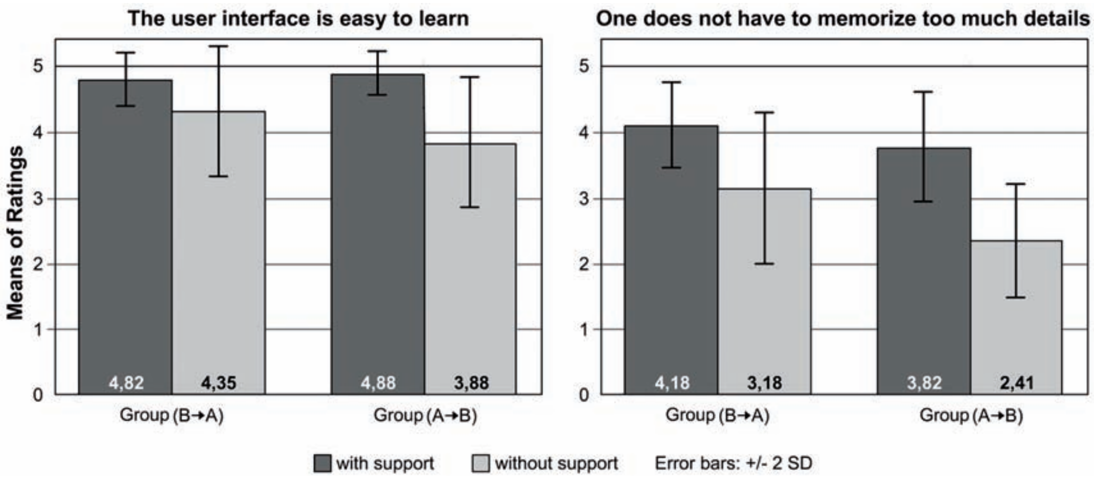


Figure 8.4: Comparison of learnability and detail memorization requirement for both variants.

ISO 9241-110 Items. Two questionnaire items were examined referring to learnability: "The user interface is easy to learn" and "The user interface does not require to memorize too many details". With respect to simplicity of learning, there was a significant main effect for workflow support, $F(1,32) = 16.23$, $p < .001$. The variant with support was better rated ($M = 4.85$, $SD = .39$) compared to without workflow support ($M = 4.12$, $SD = 1.09$). No significant effect was obtained for both condition order and interaction between support and order. Regarding the number of details to be memorized, there was a significant effect for workflow support, $F(1,32) = 29.24$, $p < .001$. Better ratings were made for workflow support ($M = 4.00$, $SD = .82$) than for no support ($M = 2.79$, $SD = 1.09$). There was also a significant effect for condition order, $F(1,32) = 5.97$, $p < .05$. In this context, Figure 8.4 shows that variant B without support was worse rated if it was executed after variant A. In contrast, variant A was better rated if performed secondly. This reveals that, when both variants were known, persons preferred the variant with support (which was also affirmed by the direct comparison described before).

Tool Processing Times. In the context of the evaluation of measurements over the time required while working with the single tools, it was discovered that working without workflow support requires less time, if one has worked with support first (cf. Figure 8.5). This leads to the assumption that workflow support has an impact on learning processes regarding the operation of single tools.

Comparisons with *Bonferroni Adjustment*[2] showed that significant less time was required when using the no support variant if groups had used the support variant first, $Diff = 1.50, p < .05$. There was no significant difference for the required time when using the support variant, depending on the order of conditions. These results verify that the provided methods of process-driven user assistance improve tool operation learning processes, considering the time required for processing.

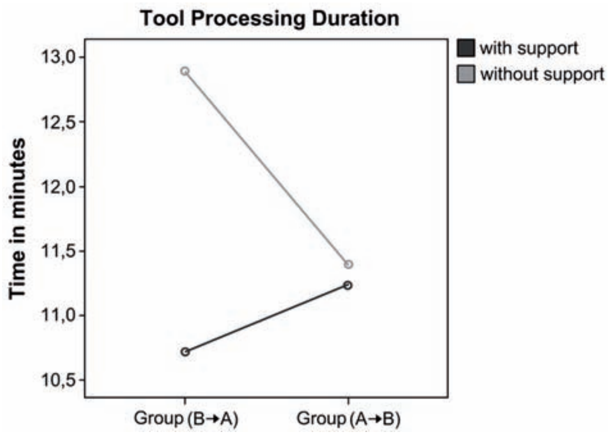


Figure 8.5: Differences of Tool Processing Times which show impact on learn effects concerning system operation.

8.2.3 Usability

For usability rating purposes, five items were included in the questionnaires, referring to the ISO NORM aspects suitability for the task, self-descriptiveness, and suitability for learning. As already explained, the respective scales were checked for reliability and subsequently mean values were calculated, before conducting a variance analysis. In comparison, a significant effect was achieved for the variant with workflow support, $F(1, 32) = 43.84$, $p < .001$. In general, the test subjects made better rankings for workflow support ($M = 4.38$, $SD = .37$) compared to variant B without support ($M = 3.48$, $SD = .78$). There was also a significant effect for the execution order, $F(1, 32) = 10.88$, $p < .05$. Figure 8.6 shows that variant B without support was worse rated if secondly applied and, in contrast, workflow support was higher rated

[2]In the scope of multiple comparisons, Bonferroni Adjustment is applied in order to keep the experimentwise α error rate to a specified level, usually .05 [Sha95].

if executed after variant B. This indicates that, as already identified for learnability, workflow support is better judged and no support is worse rated when both variants already had been processed.

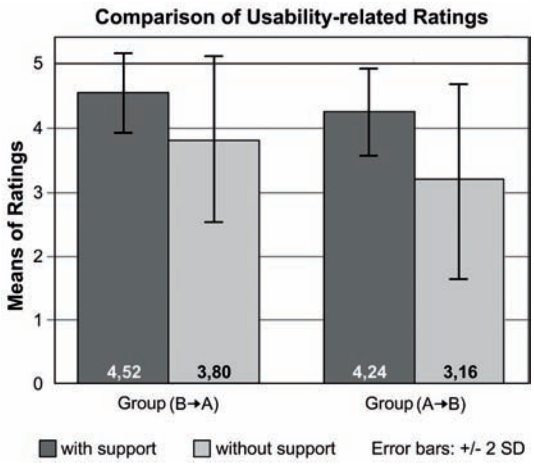


Figure 8.6: Comparison of inquired Usability-related items.

8.2.4 Process Overview

In comparison, items at part 3 of the two variant-related questionnaires requested an assessment of the obtained general process overview. These included the overview of tasks and their order, and if the UI shows (i) which is the recent task, (ii) which task have already been accomplished, and (iii) how many media files have been completed. In addition to that, the questionnaire for variant A explicitly concerned the realized concept of process-driven user assistance. As opposed to to the variant-related items, these scales were analyzed separately without previous calculation of scale mean values. Furthermore, the final questionnaire requested two free-text answers on which aspects are particularly favorable and which points need improvement, referring to the provided methods for process-driven user assistance. Finally, it was tested if errors and uncertainties occurred with respect to the selection of the next task/tool, although test subjects were provided with a paper-based description of all tasks and the the procedure to perform. In this context, it is assumed that a better overview of the process leads to a reduction of selection-related errors and uncertainties.

Comparison of the General Process Overview

As a result of summarizing scales and conducting variance analysis for comparing items, there was a significant main effect for workflow support, $F(1, 32) = 196, 28$, $p < .001$, at which workflow support was significantly better judged with $M = 4.58$, $SD = .50$ in comparison to no support with $M = 2.40$, $SD = .98$. The execution order revealed a significant effect for workflow support, $F(1,32) = 23.72$, $p < .001$. This is represented in Figure 8.7. As applies to learnability and usability, workflow support was better rated and no support worse rated when performed in the second run.

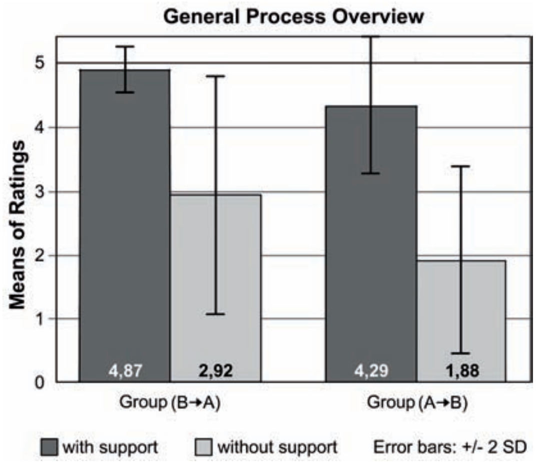


Figure 8.7: Comparison of the General Process Overview provided by both Variants.

Explicit Evaluation of Process-driven User Assistance

As described above, the realized methods for process-driven user assistance were evaluated in two ways. First, multiple choice items were applied in order to rate different partial aspects. Second, free-text comments had to be given referring to positive and negative impressions.

Multiple Choice Items. Figure 8.8 shows that all items explicitly referring to the provided workflow visualization and workflow-driven supply of tools were favorably valued. Here, inquired items addressed (i) the workflow visualization, (ii) the determination of task order, (iii) the representation of the recent task, (iv) the visualization of already accomplished tasks, (v) the visualization of already completed media documents, (vi) the automatic invocation and closing of tools. The scale was structured from 1 (strongly disagree) to 5 (strongly agree), and a summary of all items results in a mean value of $4,26$. The entire set of ratings is illustrated in Appendix D.1.

Free-text Answers. Regarding the provided methods for process-driven user assistance, free-text answers were requested concerning which aspects are particularly favorable and which points need improvement. Here, the single answers are summarized. The entire set of answers is included in Appendix D.2.

With respect to positive viewpoints, a basic statement was that the workflow visualization provides a well overview of the workflow concerning the tasks that need to

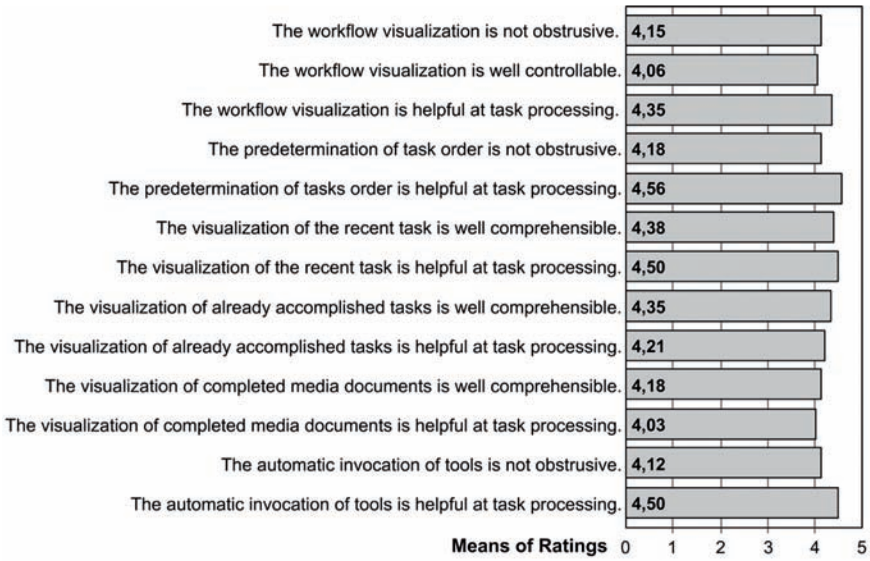


Figure 8.8: Rating of the single Aspects of Process-based User Assistance.

be performed as well as their execution order, so that the key points of the process to be conducted are quick and easy to understand. Furthermore, the automatic opening and closing of tools was positively judged. Stated reasons were that one does not require to think about which is the next task and the next tool to select, so that no procedure-related issues need to be explicitly memorized. Moreover, it was stated that the visualization of the workflow state (accomplished tasks and media documents) additionally improve the process overview, since one is permanently aware of the current position within the overall workflow. A further point mentioned multiple times was that the workflow support saves time, and less interactions and mouse clicks are required. Finally, it was pointed out that process-driven assistance allow an easy access to a new application and new tasks and requirements on operators, especially regarding novice users.

Regarding stated aspects which were found negative or which need improvement, most statements referred to general usability issues which are not primary relevant in the scope of this examination. Concerns were expressed with respect the guidance of users, at which a paternalism of users is criticized. Since the reduction of a too strong user guidance was intended and realized by the concept, a conclusion might be that more activities of free task selection could have been integrated in the study. Nevertheless, as illustrated in Table D.7 in Appendix D.1, the aspects of automatic tool supply obtained the highest ratings compared to the other items. Furthermore, one subject stated that users are not enabled to decide whether to work with or without process assistance. Actually, this aspect is realized in SemAnnot, but was not explicitly shown to test subjects in the scope of the study. Finally, two persons mentioned that graphical items and fonts within the workflow visualization panel need to be enlarged.

Errors and Uncertainties at Task/Tool selection

Errors and uncertainties were checked for acts of selection of the next task (for variant A with support) or tool (for variant B without support), assuming a significant reduction by the provision of an improved process overview.

Errors. With respect to the selection of the wrong task or tool corresponding to the determined order of task processing, there was a significant effect for workflow support, $F(1,32) = 58.31$, $p < .001$. Here, 0 errors were counted for variant A with support ($M = 0.00$, $SD = 0.00$), while 50 errors were made during the operation of the system without support ($M = 1.47$, $SD = 1.11$).

Uncertainties. Activities at tool or task transition were rated as uncertain, if the test subject halted or explicitly expressed this state. There was a significant main effect for workflow support, $F(1,32) = 122.77$, $p < .001$. Here, 2 uncertainties were registered during the operation of variant A with support ($M = 0.06$, $SD = 0.24$), at which the subjects halted after the respective tool was automatically opened. In contrast, 119 uncertainties were located while persons operated variant B without support ($M = 3.50$, $SD = 1.94$).

Concluding, this examination revealed that, although the subjects were provided with an explicit description of the tasks and procedures, still errors and uncertainties could be registered, and a significant difference between both application variants was obtained. Besides of the user-centered evaluation of the provided process overview, this indicates that the realized concepts of process-driven user assistance effectively offer an improved overview of the annotation workflow to be performed, so that enhanced mental representations can be constructed.

8.2.5 User Satisfaction

As a last resort, it was examined whether the previously analyzed viewpoints finally lead to an improvement of user satisfaction. Also here, means were created for the single scales after checking reliability.

There was a significant main effect for workflow support, $F(1,32) = 47.38$, $p < .001$. With respect to user satisfaction, workflow support obtained better ratings ($M = 4.18$, $SD = .51$) compared to the system variant without support ($M = 3.49$, $SD = .61$). As illustrated in Figure 8.9, when executed in the second run, workflow support was better rated and the ratings for variant without support became worse. Thus, workflow support led to a higher user satisfaction level, if subjects already knew the variant without support. In contrast, if variant with support had already been executed, subjects gave worse ratings for variant B. This indicates that the difference of ratings incremented, if both variants were known.

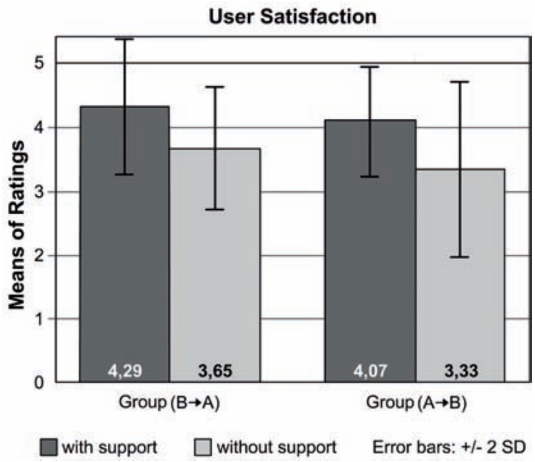


Figure 8.9: Comparison of User Satisfaction for both Variants.

8.3 Conclusions

A user study conducted by means of two different variants of SemAnnot revealed that the central contribution of this thesis - Process-driven User Assistance - *significantly* enhances the operation of multimedia annotation systems by supporting users according to the given annotation workflow. In this context, improvements on different levels were proved, which can also be regarded as key benefits achieved in this work: (i) *Efficiency*, (ii) *Learnability*, (iii) *Usability*, (iv) *Process Overview*, and (v) *User Satisfaction*.

Efficiency refers to the required time when executing an annotation process. Here, three different forms of time spaces were investigated: duration of an entire annotation run, duration of tool/task transitions (time between finishing tool A and invocation of tool B), and the time spent during operation of single annotation tools. In all three cases, a significant save of time was achieved using the variant with workflow support, obtaining significant effects both for workflow support and for the interaction between support and condition order. With respect to the condition order, the cross evaluation setting permitted the identification of further effects triggered by process assistance. In this scope, especially the time spent during operation of single annotation tools was significantly reduced, when workflow support had been performed in the first run. Thus, an improvement of learning processes was demonstrated with respect to the usage of different integrated tools.

Concerning the latter aspect, *Learnability* was measured, regarding the suitability of an user interface for learning its functions and required interactions. Here, subjects had to explicitly decide which variant is easier to learn, at which 85.3% of the respondents opted for workflow support. Furthermore, the ISO NORM items “The user interface is easy to learn” and “The user interface does not require to memorize too many details”were inquired for each variant. Here, significant better ratings were given for workflow support. Moreover, a significant effect appeared for execution order, showing that variant A with support is better ranked and variant B without support is worse rated, when both variants already had been performed. As stated above, further learn effects could be demonstrated which refer to the operation of single annotation tools integrated in a given workflow, indicated by significatly reduced tool processing times.

For *Usability* rating purposes, five items were included in the questionnaires, referring to the ISO NORM aspects suitability for the task, self-descriptiveness, and suitability for learning. Here, variant A with process-driven assistance was significantly better valued. Here too, when executed in the second run, variant with support was better rated and variant without support worse rated, compared to the results when performed in the first run.

The provided *Overview of the Annotation Process*, regarded as ability of the UI to allow users to construct an appropriate mental representation of the given workflow, was measured in three ways. First, multiple choice items were included in both variant-related questionnaires, allowing a comparison of the general process overview. Here, variant A with support was significantly better valuated. As already identified for learnability and usability, better ratings for A and worse ratings for B were given when both variants already had been accomplished. Furthermore, the methods for process-driven user assistance provided by variant A were inquired, including (i) the workflow visualization, (ii) the determination of task order, (iii) the representation of the recent task, (iv) the visualization of already accomplished tasks, (v) the visualization of already completed media documents, (vi) the automatic invocation and closing of tools. Here, all items obtained good ratings with a mean of 4.26 at a scale from 1 to 5. Additionally, errors and uncertainties at transition between different tasks or tools were examined, assuming that a better process overview would reduce the number. For both mistakes and uncertainties, a significant reduction was achieved with workflow support (although subjects obtained a detailed paper-based task description). Concluding, it was proven that a significant improvement of the overview and respective mental representations of the given workflow was achieved through the application of process-driven user assistance.

Furthermore, it was demonstrated that process assistance leads to a significantly higher degree of *User Satisfaction*, potentially induced by the improvements obtained regarding the previous viewpoints.

Finally, the following effect was noticed with respect to user-centered ratings commonly requested for both variants, particularly respecting usability, general process support, and user satisfaction: group α performing first the variant with support and second variant without support, gave better ratings for A and worse ratings for B, compared to the valuation of variants by group β. This revealed that in direct comparison, that is, when both variants were known, the significance of the difference between ratings incremented. In other words, when the subjects had already performed both variants, the significance of the rating variance increased.

To sum up, the user study showed that the application of the proposed methods for process-driven user assistance leads to a significant improvement of system operation during the execution of multimedia annotation processes.

Chapter 9

Conclusions

Practices of digital multimedia annotation have found their way into multiple areas of daily use and professional work. This includes a variety of different application purposes and domains, as well as a multitude of supported media and annotation formats, which is particularly identifiable in recent multimedia annotation systems. Consequently, respective graphical user interfaces are to be characterized as complex, providing a diversity of tools which allow the execution of several different annotation tasks. This leads to an aggravation of the annotation process, at which users often struggle with decisions on which is the current task to perform and which is or are the correct applicable tool(s) to invoke.

Correspondingly, the general objective of this thesis was to establish workflow support by providing visual and interactive access to all tasks to perform, their sequences, and relevant information about the workflow state. Additionally, in the scope of an automatic workflow execution, this was to be accompanied by an explicit supply of the suitable annotation tools or services, depending on the recent task to perform. Moreover, since no best practices of annotation can be identified due to the described manifoldness of this domain, a further goal was to contribute a method of individually specifying annotation workflows by means of standard modeling techniques. As additional aspect, also collaborative workflows of multimedia annotation were taken into account.

This chapter summarizes the main contributions of this thesis in Section 9.1. Section 9.2 illustrates the benefits related to system operation, which were achieved by the application of process-driven user assistance. Finally, prospects for directions of future research are provided in Section 9.3.

9.1 Summary

This thesis contributes to the human-centered execution of (collaborative) multimedia annotation processes. The partial aspects will be discussed in the following sections.

9.1.1 Initial Studies on Multimedia Annotation

As initial step, three different studies on multimedia annotation were conducted, establishing new outcomes with respect to related work. First, an *Analysis of Features* provided by multimedia annotation systems was illustrated in Chapter 3, including functionalities, tools, and approaches offered in order to accomplish different forms of annotation tasks. In addition to that, general forms of data were identified which are generated and treated by annotation-related features. In this context, the problems and challenges defined in the specific scope of this thesis could be affirmed, revealing that annotation systems (and respective UIs) provide a diversity of functionalities and supported media and annotation formats. In addition to that, a set of annotation features was developed and assigned to general task areas of annotation, based on an inductive development of activity-related categories. According to this, annotation features were associated to activities of (i) system configuration, (ii) area selection, (iii) information attachment, (iv) search and exploration, (v) and import and export. Moreover, specific functionalities were localized which are incorporated for collaboration and process support. With regard to the latter point, it was shown that existing work in fact does not sufficiently consider an assistance of annotation workflows.

Second, studies on annotation practices were conducted, including an *Empirical Study* and a *Survey of Related Literature* (Chapter 4). The empirical study involved experts at five research and educational institutes, which were interviewed and observed while operating the respectively applied annotation system. The conducted literature survey summarized existing work on annotation workflow models, field reports, and project procedure proposals. As a result, the annotation process was subdivided into enclosed activity clusters, and relations between these clusters were identified with regard to their operational procedures and execution orders. Furthermore, the studies revealed that no best practices of annotation can be identified, confirming the specified need for an individual modeling of annotation workflows.

9.1.2 Generic Process Model of Multimedia Annotation

Based on the results of the initial studies on multimedia annotation, a *Generic Process Model of Multimedia Annotation* was established in Chapter 5, generating fundamental knowledge about the activities and procedures associated with multimedia annotation. The model was constructed in abstract manner, covering different use cases and workflows of annotation of varying media formats and thus exceeding existing models, also due to the additional consideration of collaborative work scenarios. In general, the following issues related to annotation processes were defined and correlated: (i) phases and sub-processes of annotation, (ii) sequences of these phases and sub-processes, also respecting jumps and iterations, (iii) single tasks performed within sub-processes, (iv) approaches provided by systems in order to accomplish specific annotation tasks, and (v) forms of data generated and processed by offered approaches.

9.1.3 Data Modeling and Management Concept

In the context of a data modeling and management concept, a *Formal Specification of Annotation Processes* was constructed in Section 6.2. Assuming that workflow management systems are driven by a specific process model, a workflow scheme was formally specified. Here, basic elements included in annotation processes were defined, considering specific properties and relations to other elements. In doing so, relevant classes of process-related information entities were declared, which need to be processed, visualized, and made accessible for users by a process-based annotation system. Among these basic process elements are (i) work items to be processed, (ii) multimedia contents and annotations handled during the process, (iii) specific conditions which determine the workflow procedure, (iv) the services assigned to work items, and (v) the human workflow participants which perform the entire annotation process or parts of it.

In addition to the declaration of workflow entities which are processed by a process-based annotation system, the formal specification serves as framework of guidelines which can be employed in the scope of workflow modeling processes, also defining general rules for workload distribution. In other words, this contribution describes how workflows of (collaborative) multimedia annotation effectively are to be specified.

9.1.4 Architectural Concept

Through the specification of an *Reference Architecture Model* (Section 6.3), the organizational structure and behavior of a process-based annotation system was defined. An arrangement by a client-server model permitted the incorporation of a multimedia annotation server, which provides functions for data exchange and sharing, data consistency, correct data handling, and user management. Additionally, a conceptualized fat-client application assumes essential features for workflow execution and monitoring. Here, a process engine composed by a workflow handler and a mediator instance was established, including specific functions for workflow interpretation and execution, as well as integration, coordination, and synchronization of incorporated annotation services and further architecture components. In addition to that, the architecture was defined as component-based environment, so that different sub-processes of annotation could be enclosed as functional units with common properties. In this context, different forms of components were introduced, defining general interfaces and interactions with the overall framework. Thus, these components represent the different services or tools integrated in a predefined annotation workflow. With regard to data junction, identified as key requirement for annotation systems in which varying types of data are generated and processed by different services, a further architectural solution was introduced. Here, component interfaces were assigned to specific modules provided by the mediator instance, which have read and write access on parts of an existing data model.

9.1.5 Visual-interactive Concept

This thesis approached user-centered problems which result from an insufficient consideration of the annotation workflow by existing work. Correspondingly, regarded as central contribution, a visual-interactive concept for *Process-driven User Assistance* was developed in Section 6.4, which describes the design and behavior of UIs of process-based systems. First of all, a 4 View Model was introduced as part of a basic visual design, which subdivides the user interface into four placement areas for different forms of components, distinguishing between elements for traditional tool selection, workflow visualization, as well as services which are required permanently or temporally in the course of the process. Furthermore, a graph-based model for interactive workflow visualization was established. In this context, basic graphical elements were defined, which represent different entities of an annotation workflow. Additionally, prevention of a too strong guidance of users was achieved by realizing two different modes of workflow support, and providing facilities for a manual selection of annotation tasks, if they are legal at the current workflow state. With respect to the latter aspect, an approach for workflow progress monitoring was established, considering the advancement of a given workload during process time and respective modifications of the system, that are also provided to users as workflow-related feedback. Finally, by means of workflow-driven service supply on behalf of the graphical user interface, users are explicitly provided with the right tool(s) or services(s), depending on the recent annotation task to be performed.

9.2 Benefits of Process Assistance

Chapter 8 illustrated a *User Study* conducted by means of SemAnnot, a framework and toolset which implements the concepts established in this thesis (cf. Chapter 7). Based on a comparison of two application variants (with and without workflow support), it was revealed that an application of process-driven user assistance significantly improves the interactive operation of multimedia annotation systems. These improvements are associated to the partial aspects *Efficiency*, *Learnability*, *Usability*, *Process Overview*, and *User Satisfaction*.

Efficiency. Referring to the required time when executing an annotation process, a significant save of time was achieved with workflow support for the three points (i) entire annotation run, (ii) duration of tool/task transitions (time between finishing tool A and invocation of tool B), and (iii) the time spent during operation of single annotation tools. Significant effects both for workflow support and for the interaction between support and execution order were obtained. In this context, detected significant effects regarding the time spent during operation of single annotation tools demonstrated that also learning processes regarding the usage of different integrated tools have been improved.

Learnability. The suitability of the user interface for learning its functions and required interactions was measured in two ways. First, subjects had to explicitly decide which variant is easier to learn, at which 85.3 % of the respondents opted for workflow support. Second, in the scope of variant-related questionnaire items, significant better ratings were given for workflow support. As stated above, further learn effects could be demonstrated which refer to the operation of single annotation tools integrated in a given workflow, indicated by significatly reduced tool processing times.

Usability. For usability rating purposes, ISO NORM questionnaires were applied which refer to (i) suitability for the task, (ii) self-descriptiveness, and (iii) suitability for learning. Here, the variant with workflow support was significantly better valued. Furthermore, when executed in the second run, variant with support was better rated and variant without support worse rated, compared to the results when performed in the first run. This revealed that the obtained difference of ratings was higher when the test subjects already knew both application variants.

Process Overview. It was proven that the provided overview of the annotation process, regarded as ability of the UI to allow users to construct an appropriate mental representation of the given workflow, was significantly improved by the application of process support. This was evaluated in three ways. First, in the context of a comparison of the general process overview within variant-related questionnaires, variant with support was significantly better valuated. Second, the concepts for process-driven user assistance established in this thesis and implemented in the first application variant were inquired, at which all included aspects obtained good ratings with a mean of 4.26 at a scale from 1 to 5. Third, errors and uncertainties at transitions between different tasks or tools were examined, assuming that a better process overview would reduce the number. Although subjects obtained a detailed paper-based task description, a significant reduction was achieved with workflow support for both mistakes and uncertainties.

User Satisfaction. It was demonstrated that process assistance leads to a significantly higher degree of user satisfaction, potentially induced by the improvements obtained regarding the previous viewpoints.

To sum up, benefits established by process-driven user assistance for operators of multimedia annotation systems were achieved with respect to a significant improvement of the following partial aspects: (i) saving of time at task processing, (ii) facilitated learning of UI functions and interactions, (iii) improved usability as measured by suitability for the task, self-descriptiveness, and learnability, (iv) sophisticated overview of the process including status changes on run-time, an (v) enhancement of user satisfaction.

9.3 Prospects for Future Work

Several directions of future research can be pointed out referring to different points, which either were defined as open issues and not pursued in greater depth, or emerged in the course of this thesis.

9.3.1 Service Description

It was illustrated that, due to an integration of several annotation services and approaches, different heterogeneous forms of generated and processed data require specific management and storage strategies. This thesis presented a solution concept on an architectural and structural level, defining the relations between elements of the system architecture and the relevant data model entities. Nevertheless, no statements were given about concrete assignment strategies for these different data items, leading to two problem factors.

First, situations have to be considered in which existing annotation data is recalled, e.g., for re-editing purposes. Problems arise from the fact that requested data needs to be represented by the correct UI component (regarded as representative of a service). To give an example, a workflow may include two different services which generate the same type of information, for instance, a video event detection and a scene detection service which both produce time intervals. In this case, a later recognition of the respective UI component which is just based on the given data entities is not trivial. Second, in the scope of flexible and extensible systems, it must be possible to replace services, for example by new versions providing improved algorithms. Here too, a solution for correct service recognition and assignment is required.

As a result, a consistent and machine-readable description of integrated services needs to be established. Related problem areas are addressed by research work conducted in the field of *service description* [LADM08]. Thus, approaches can be referenced that are associated, among others, to *service composition* or *semantic service modeling*. For instance, a semantic annotation of services can foster service discovery as well as their composition into workflows [BEP+08, MK10]. Hence, existing approaches need to be directed towards the specific properties and requirements of functions and data related to digital multimedia annotation.

9.3.2 Change Tracking

In Chapter 5, it was pointed out that annotation workflows often are not pipeline-like, but consist of constant returns to previous tasks and loops, so that the final results arise from iterative editing processes. Such iterations, implying proofs and data modifications, are mostly initiated by explorative activities which accompany acts of area selection and information addition, at which existing results are reviewed and compared. In the scope of collaborative work settings, also data generated by co-annotators plays a role, particularly changes and updates on parts of the data set which are commonly processed. Consequently, relevant future research considers tracking of changes conducted by other users at run-time or periods of absence. The basic principles of change tracking in collaborative workspaces are described in [TG06].

In the context of change tracking for collaborative annotation scenarios, few existing work can be identified up to now. In the scope of conducted research work associated with this thesis, an approach referring to the specific challenges of video-based media was addressed in Hofmann et al. [HBF10], focusing on bounding boxes that arise from video object tracking. Moreover, Papadopoulou [Pap09] presented a framework that supports change awareness for collaborative text authoring work settings. Nevertheless, existing work only deals with one or single aspects of multimedia annotation. Consequently, comprehensive research work is still lacking, considering the specific challenges and requirements of different multimedia formats, forms of selection areas, and annotation types.

9.3.3 Machine Learning

The research area of machine learning bears approaches in the scope of different application domains, aiming at the establishment of (artificial) intelligence in order to enable computers to evolve their behavior, based on inputs or empirical data, or in response to external information. Supported tasks include recognition, planning, or prediction.

With respect to processes of multimedia annotation, specific machine learning approaches can be applied to modify existing workflow specifications during a running process, depending on the activities performed by operators. For instance, this can be realized by the integration of interaction analysis and corresponding prediction calculations [OL10].

It can be expected that the application of machine learning approaches brings benefits particularly for annotation workflows, which imply a high degree of freedom respecting self-motivated or requested manual selection of tasks, including frequent points of user-centered decision about the further workflow procedure.

9.3.4 Workflow Modeling

Finally, discussions held with experts both in the course of the conducted empirical study as well as at attended conferences revealed that the planning, conception, and configuration of an annotation project still is an elaborate and complex process.

In the context of workflow modeling, this refers to the assignment of different information entities, including tasks, transitions, conditions, services, data formats, and involved users. For instance, a workflow modeler still needs to acquire knowledge about concrete implementation details in order to address specific interfaces or code parts in the scope of service assignment. Thus, corresponding authoring facilities are required, which provide a sufficient degree of abstraction for different forms of given information, and support several types of interaction connected to modeling processes.

Appendix A

Listing and Classification of Annotation Systems

Open Standards	
Annotea	http://www.w3.org/2001/Annotea/
Time-based metadata standard draft	http://annodex.net/TR/draft-pfeiffer-temporal-fragments-03.html
W3C Media Fragments WG	http://www.w3.org/2008/WebVideo/Fragments/
W3C Media Annotation WG	http://www.w3.org/2008/WebVideo/Annotations/
Digital Document Annotation	
Adobe Acrobat	www.adobe.com/de/products/acrobat/
A.nnotate	http://a.nnotate.com/
COLLATE	http://www.kulturerbe-digital.de/de/projekte/9_38_343634.php
CoScribe	http://www.tk.informatik.tu-darmstadt.de/index.php?id=998
DILAS	http://www.is.informatik.uni-duisburg.de/bib/docs/ Agosti_etal_05.html
Foxit Reader	foxit-pdf-reader.softonic.de/
PDFedit	http://pdfedit.petricek.net/en/index.html
Vannotea	http://www.itee.uq.edu.au/ eresearch/projects/vannotea/index.html
WebNotes	http://www.webnotes.net/
XLibris	http://www.fxpal.com/?p=xlibris
Web Annotation	
Diigo	http://www.diigo.com/
DrawHere	http://drawhere.com/
Marginalia	http://code.google.com/p/marginalia/

MADCOW	http://www.web-notes.com/index.php
MyStickies	http://www.mystickies.com/
Pronotes	http://www.protonotes.com/
SharedCopy	http://sharedcopy.com/
ShiftSpace	http://www.shiftspace.org/
Stickis	http://stickis.com/
TrailFire	http://trailfire.com/
WebAnn	http://doi.acm.org/10.1145/506443.506610
Text Annotation	
Commentary	http://pythonpaste.org/commentary/
Django's Comment System	http://www.djangobook.com/
eMargo	emargo.de/www/start/index.html
Stet	http://en.wikipedia.org/wiki/Stet_(software)
Image Annotation	
ALIPR	http://www.alipr.com/
behold	http://www.behold.cc/
EasyAlbum	http://research.microsoft.com/en-us/groups/vc/easyalbumdownload.aspx
Flickr	http://www.flickr.com
Google Image Labeler	http://images.google.com/imagelabeler/
Omeka Image Annotation Plugin	http://omeka.org/codex/Plugins/ImageAnnotation
Open Layers	http://openlayers.org/
PhotoStuff	http://www.mindswap.org/2003/PhotoStuff
SpiritTagger	http://cortina.ece.ucsb.edu/index.php/
Audio Annotation	
BBC AAP	http://www.plasticbag.org/archives/2005/10/on_the_bbc_annotatable_audio_project/
Music Annotator	http://clam-project.org/wiki/Music_Annotator
Project Pad	http://dewey.at.northwestern.edu/ppad2/documents/help/audio.html
Sonic Visualiser	http://www.sonicvisualiser.org/
WaveSurfer	http://www.speech.kth.se/wavesurfer
Video Annotation	
Adivi	http://www.adivi.de
BubblePLY	http://www.bubbleply.com
bvault	https://launchpad.net/bvault
eSports	http://grids.ucs.indiana.edu/ptliupages/publications/

	eSportsFinalDSpace.pdf
Kaltura	http://corp.kaltura.com/
Metavid	http://metavid.org/wiki/
MRAS	http://www.francisli.com/portfolio/mras.html
Pad.Ma	http://pad.ma/
Youtube	http://www.youtube.com
VARS	http://www.mbari.org/vars/
Viddler	http://www.viddler.com
VideoAnnEx	http://www.research.ibm.com/VideoAnnEx/
VITAL	http://ccnmtl.columbia.edu/our_services/tools/vital/
3D-Model Annotation	
3D-Tool	http://www.3d-tool.de/english/cad-viewer.htm
AnnoCryst	http://www.itee.uq.edu.au/ eresearch/projects/dart/outcomes/ crystallography.php
CATIA 3D	http://www-01.ibm.com/software/applications/plm/catiav5/ prods/fta/
WorkXPlore 3D	http://www.workxplore-3d.com/uk/product/functions/ annotations/
Social Annotation & Bookmarking	
Bibsonomy	www.bibsonomy.org/
citeulike	www.citeulike.org
Connotea	http://www.connotea.org/
Delicious	delicious.com/
LAST.fm	www.last.fm/
Pandora	http://www.pandora.com
Semantic Annotation	
Amaya	http://www.w3.org/2001/Amaya
Anozilla	http://annozilla.mozdev.org/
COHSE	http://cohse.semanticweb.org/
EVA	http://domino.watson.ibm.com/comm/research.nsf/pages/ r.multimedia.innovation.html
KIM	http://www.ontotext.com/kim/semanticannotation.html
Mangrove	http://www.cs.washington.edu/research/semweb/
MnM	http://www.aktors.org/technologies/mnm/
M-Ontomat-Annotizer	http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html
OntoMat-Annotizer	http://annotation.semanticweb.org/ontomat/index.html
Open Ontology Forge	http://research.nii.ac.jp/ collier/resources/OOF/index.htm
SMORE	http://www.mindswap.org/2005/SMORE/

Content Analysis Annotation	
Anvil	http://www.anvil-software.com
ATLAS.ti	http://www.atlasti.com
Interact	http://www.mangold-international.com/de/produkte/software/interact.html
The Observer	http://www.noldus.com/office/de/observer-xt-an
Transana	http://www.transana.org/
Videograph	http://www.ipn.uni-kiel.de/aktuell/videograph/htmStart.htm
v-share	http://www.v-share.de/
WebDiver	http://diver.stanford.edu/webdiver.html
Audio and Video Transcription Annotation	
Annotation Graph Toolkit	http://agtk.sourceforge.net/
ELAN	http://www.lat-mpi.eu/tools/elan/
EXMARaLDA	www.exmaralda.org/
Praat	www.fon.hum.uva.nl/praat/
Shoebox	http://www.sil.org/computing/shoebox/TipIndex.html
TASX	http://annotation.exmaralda.org/index.php/TASX

Table A.1: List of Multimedia Annotation Systems.

.

Appendix B

Empirical Study: Questionnaire on (Collaborative) Practices of Multimedia Annotation

Erhebung zu Nutzung und Ablauf von (kollaborativer) Multimedia Annotation

Angaben zum Institut

Teilnehmende Personen

Fragen

1. Allgemeiner Kontext

- In welchem Rahmen wird Annotation durchgeführt? (z.B. Hochschulseminare, Trainings, etc.)
- Falls Lernszenario: Unterliegt der Annotationsprozess einem besonderen didaktischen Design?
- Bezogen auf „Wissenserwerb" / "Erkenntnisgewinn": Welche Ziele bzw. welchen Zweck hat der Einsatz von Annotation in Ihrer Veranstaltung? (z.B. Bewertung von gefilmten Bewegungen, Erlernen von Argumentationsstrategien, Identifikation von Interaktionsmustern, etc.)
- Wird Annotation in einem kollaborativen Gruppenszenario durchgeführt? Wenn ja → Punkt 4.
- Existieren besondere/ relevante Gesichtspunkte, die hier bisher nicht erhoben wurden?

1/4

2. Annotationsprozess

2.1. Vor dem „eigentlichen“ Annotationsprozess

- Welche Handlungen gehen dem eigentlichen Annotationsprozess voraus? (z.B. Daten aufzeichnen, hochladen und bereitstellen, Benutzer und Gruppen definieren, Kategorien/Vokabular definieren, etc.)
- Wie hoch ist der organisatorische Aufwand?
- Was gehört zum organisatorischen Teil?
- Existieren besondere/ relevante Gesichtspunkte, die hier bisher nicht erhoben wurden?

2.2. (Kollaborativer) Annotationsprozess

- In welchem zeitlichen Rahmen wird Annotation durchgeführt?
- Können Sie den Ablauf des Annotationsprozesses beschreiben?
- Kann man den Gesamtprozess in unterschiedliche Phasen einteilen? (z.B. Exploration, Festlegung relevanter Inhalte, annotieren, diskutieren, etc.)
- Wenn Phasen zu identifizieren sind: Können diese Phasen in einzelne Teilprozesse bzw. Arbeitsschritte „aufgebrochen“ werden? (z.B. a. konkreten Inhalt markieren, b. Inhalt verschlagworten/ kategorisieren, c. Annotation eingeben)
- Gibt es bestimmte (feste) Reihenfolgen,?
- Welche Funktionalitäten werden in den Arbeitschritten benutzt?
- Welche Daten werden in den Arbeitsschritten erzeugt?
- Existieren besondere/ relevante Gesichtspunkte, die hier bisher nicht erhoben wurden?

2.3. Nach dem „eigentlichen" Annotationsprozess

- Welche Handlungen geschehen nach dem eigentlichen Annotationsprozess?
- Gibt es bestimmte Ausgabedokumente (z.B. Print) oder andere Speicherformate (z.B. HTML), in denen Ergebnisse zusammengefasst werden?
- Werden Projekte und ihre Daten archiviert/ gelöscht/ bei nachfolgenden Projekten berücksichtigt?
  - Wenn berücksichtigt: Wie? Wird das durch die Software unterstützt?
- Existieren besondere/ relevante Gesichtspunkte, die hier bisher nicht erhoben wurden?

3. Software Unterstützung

- Wird eine spezielle Annotationssoftware verwendet?
- Wenn ja: Welche?
- Wenn ja: Welche Funktionalitäten oder Tools bietet die Software an?
- Auf welche Art erlernen Anwender die Benutzung der Software? (z.B. Vorab-Trainigs, autonome Übungsphasen, Tutorials, etc.)
- Wie schätzen Sie die Zufriedenheit der Anwender mit der Software ein?
- Treten spezielle Probleme bei der Nutzung der Software auf?
- Werden andere Applikationen zur Unterstützung der Annotation von Mediendokumenten benutzt? (z.B. Texteditoren, Graphikprogramme, spezielle Analysewerkzeuge, Email, Chat, etc.)

- Gibt es Phasen oder Arbeitsschritte, in denen keine Computerunterstützung angewendet wird? (z.B. Papier & Stift, Diskussion in Präsenzveranstaltungen, etc.)

- Existieren besondere/ relevante Gesichtspunkte, die hier bisher nicht erhoben wurden?

4. Kollaboration

- Handelt es sich um synchrone oder asynchrone Kollaboration?

- Welche Personen und Rollen sind am kollaborativen Prozess beteiligt? (z.B. Lehrende, Lernende, Trainer, Wissenschaftler, Administrator, etc.)

- Welchen Rollen werden welche Rechte bzw. Befugnisse zugeteilt?

- Treten - bei der Nutzung der Software oder auch generell - spezifische Probleme auf, die dem kollaborativen Prozess zuzuordnen sind?

- Werden die Ergebnisse der kollaborativen Arbeit in irgendeiner Weise „zusammengefasst“?

- Wann genau wird kollaborativ, wann wird individuell gearbeitet?

- Existieren besondere/ relevante Gesichtspunkte, die hier bisher nicht erhoben wurden?

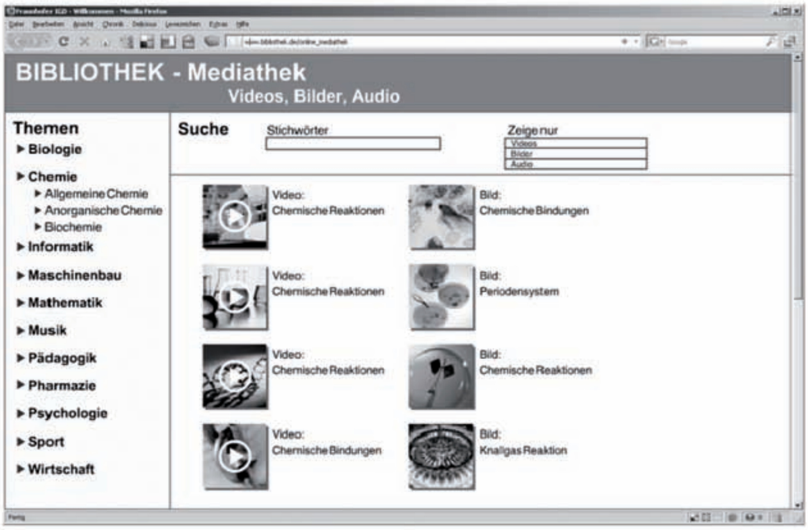
Appendix C

User Study: Additives and Questionnaires

Evaluation der Software für Multimedia Annotation (Variante A: mit Prozessunterstützung)

BASIS SZENARIO

Du bist Mitarbeiter an einer Bibliothek, welche neben Fachbüchern auch ein Online-Portal anbietet, in dem Multimedia-Dokumente wie z.B. Videos oder Bilder zur Verfügung gestellt werden. Diese Medien werden nach bestimmten Themen- und Fachgebieten sortiert, und können so mittels einer speziellen Navigationsleiste und einer Suchmaschine abgerufen werden.



AUFGABE

Zum Thema „Grundarten von Chemischen Reaktionen" sind 3 neue Videos und 2 neue Bilder eingegangen. Deine Aufgabe ist es nun, die neuen Medieninhalte thematisch einzuordnen, um diese im Online-Portal zur Navigation und Suche bereitstellen zu können. Dazu musst Du die zur Verfügung stehende Software zur Annotation von Multimedia Dokumenten verwenden. Führe dabei die folgenden Aufgaben aus (wir nehmen an, das Programm ist bereits geöffnet).

1. Auswahl der Mediendokumente

Bevor Du die Annotation starten kannst, müssen zunächst die 3 Videos und 2 Bilder geöffnet werden. Wähle die Datenbank „Chemie Videos und Graphiken" und selektiere darin die folgenden Medien:

a. Bild: Faellungsreaktion Silbernitrat Natriumnitrat
b. Bild: Quecksilberoxid Zerlegung
c. Video: Chemische Stoffe: Amoniak Molekuel
d. Video: Chemical Reactions: Synthesis and Decomposition
e. Video: Chemical Reactions: Single and Double Displacement

Führe die Schritte 2 - 4 nun für jedes Dokument nacheinander aus.

2. Auswahl der Bereiche, die kategorisiert werden

In einem Video oder einem Bild können auch einzelne Teile enthalten sein, die einem untergeordneten Thema zugehören. Beispiel: Ein Video handelt über „Grundarten Chemischer Bindungen". Von Sekunde 5 – 20 wird die „Ionische Bindung" und von Sekunde 60 – 120 wird „Kovalente Bindung" erläutert. Somit ist nun Deine Aufgabe, solche Bereiche zu markieren.

Für Videos wird das Werkzeug „Berechnung Kameraschnitte" und für Bilder das Werkzeug „Markierung Bildbereiche" eingeblendet.

3. Kategorisierung durch Stichwörter

Damit die späteren Portalbesucher Medien bzw. Teile der Medien mit Hilfe der Suchmaschine finden können, müssen die Stichwörter definiert und angeknüpft werden, die die passenden Themengebiete beschreiben.

Hierzu wird das Werkzeug „Verschlagwortung von Medienbereichen" eingeblendet. Hierbei ist es Deine Aufgabe, zunächst das ganze Video oder Bild mit Stichwörtern zu kategorisieren, und anschließend die zuvor definierten Teilbereiche.

4. Einordnung in die Themenstruktur

Die Mediendokumente können nun direkt in die Themenstruktur eingegliedert welchen, die auf der Website durch die Navigationsleiste dargestellt wird.

Hier wird das Werkzeug „Einordnung in Semantik" eingeblendet. Dieser visualisiert die gegebenen Themen. Nun musst Du vom „Annotation Browser" ganze Medien oder Medienteile durch *Drag&Drop* in die Elemente der visuellen Darstellung ziehen.

Solange Du noch nicht alle Medien fertig bearbeitet hast, gehe zurück zu Schritt 2.

5. Zusatzaufgabe: Fehlerkorrektur

Am Ende des Arbeitsprozesses stellst Du fest, dass Dir bei der Kategorisierung von Video „Chemical Reactions: Synthesis and Decomposition" ein Rechtschreibfehler unterlaufen ist.

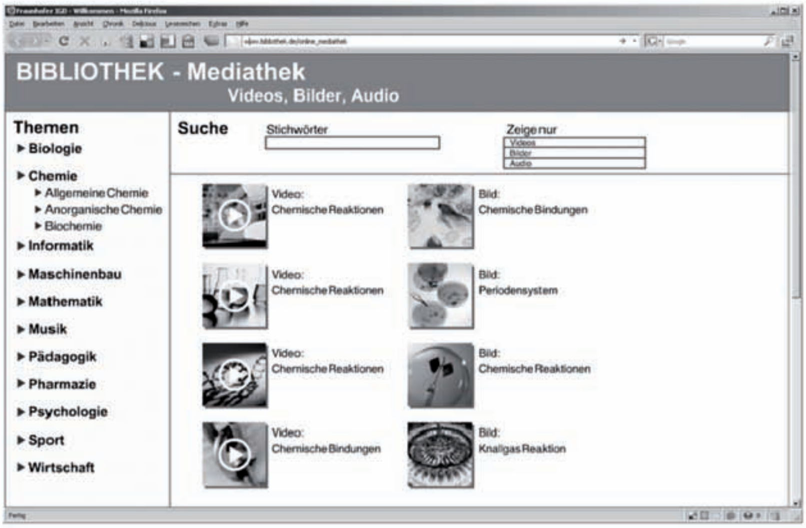
Um dies zu korrigieren, musst Du das Video nochmals aufrufen. Dies musst Du mit dem mit „Lade bereits annotierte Dokumente" tun. Wähle nun das Video und führe nochmals „Definiere Schlagwörter" aus. Hier kannst Du nun den alten Eintrag überschreiben. Mache das bitte für den Teilbereich „00:14-00:27"

2 / 2

Evaluation der Software für Multimedia Annotation (Variante B: ohne Prozessunterstützung)

BASIS SZENARIO

Du bist Mitarbeiter an einer Bibliothek, welche neben Fachbüchern auch ein Online-Portal anbietet, in dem Multimedia-Dokumente wie z.B. Videos oder Bilder zur Verfügung gestellt werden. Diese Medien werden nach bestimmten Themen- und Fachgebieten sortiert, und können so mittels einer speziellen Navigationsleiste und einer Suchmaschine abgerufen werden.



AUFGABE

Zum Thema „Grundarten von Chemischen Reaktionen" sind 3 neue Videos und 2 neue Bilder eingegangen. Deine Aufgabe ist es nun, die neuen Medieninhalte thematisch einzuordnen, um diese im Online-Portal zur Navigation und Suche bereitstellen zu können. Dazu musst Du die zur Verfügung stehende Software zur Annotation von Multimedia Dokumenten verwenden. Führe dabei die folgenden Aufgaben aus (wir nehmen an, das Programm ist bereits geöffnet).

1. Auswahl der Mediendokumente

Bevor Du die Annotation starten kannst, muss zunächst das Video oder Bild, das in einem Durchlauf annotiert wird, ausgewählt werden. Starte dazu das Werkzeug *„Suche in Datenbank"*. Wähle nun die Datenbank „Chemie Videos und Graphiken" und selektiere darin bei jedem neuen Durchlauf eine der folgenden Medien:

a. Video: Classes of Chemical Reactions: Double Displacement Combustion
b. Video: Classes of Chemical Reactions: Synthesis Decomposition
c. Video: Chemische Stoffe: Stickstoff Molekuel
d. Bild: Stoffzerlegung von Quecksilberoxid 2
e. Bild: Synthese von Kupfer Schwefel

2. Auswahl der Bereiche, die kategorisiert werden

In einem Video oder einem Bild können auch einzelne Teile enthalten sein, die einem untergeordneten Thema zugehören. Beispiel: Ein Video handelt über „Grundarten Chemischer Bindungen". Von Sekunde 5 – 20 wird die „Ionische Bindung" und von Sekunde 60 – 120 wird „Kovalente Bindung" erläutert. Somit ist nun Deine Aufgabe, solche Bereiche zu markieren.

Für Videos musst Du dazu das Werkzeug „Berechnung Kameraschnitte" und für Bilder das Werkzeug „Markierung Bildbereiche" benutzen.

3. Kategorisierung durch Stichwörter

Damit die späteren Portalbesucher Medien bzw. Teile der Medien mit Hilfe der Suchmaschine finden können, müssen die Stichwörter definiert und angeknüpft werden, die die passenden Themengebiete beschreiben.

Verwende dazu das Werkzeug „Verschlagwortung von Medienbereichen". Hierbei ist es Deine Aufgabe, zunächst das ganze Video oder Bild mit Stichwörtern zu kategorisieren, und anschließend die zuvor definierten Teilbereiche.

4. Einordnung in die Themenstruktur

Die Mediendokumente können nun direkt in die Themenstruktur eingegliedert welchen, die auf der Website durch die Navigationsleiste dargestellt wird.

Rufe dazu das Werkzeug „Einordnung in Semantik" auf. Dieser visualisiert die gegebenen Themen. Nun musst Du vom „Annotation Browser" ganze Medien oder Medienteile durch *Drag&Drop* in die Elemente der visuellen Darstellung ziehen.

Solange Du noch nicht alle Medien fertig bearbeitet hast, gehe zurück zu Schritt 1.

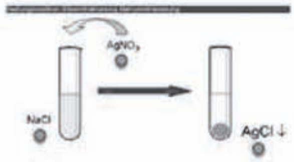
5. Zusatzaufgabe: Fehlerkorrektur

Am Ende des Arbeitsprozesses stellst Du fest, dass Dir bei der Kategorisierung von Video „Classes of Chemical Reactions: Synthesis Decomposition" ein Rechtschreibfehler unterlaufen ist.

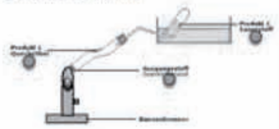
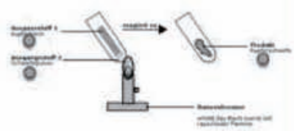
Um dies zu korrigieren, musst Du das Video nochmals aufrufen. Dies musst Du mit dem Werkzeug „Lade bereits annotierte Dokumente" tun. Wähle nun das Video und führe nochmals „Definiere Schlagwörter" aus. Hier kannst Du nun den alten Eintrag überschreiben. Mache dies bitte für den Teilbereich „00:13-00:25"

2 / 2

Evaluation der Software für Multimedia Annotation (Variante A: mit Prozessunterstützung)

Medien (Name)	Abschnitt	Schlagwörter	Einordnung: Thema
VIDEO: Chemical Reactions: Single and Double Displacement	**GANZES VIDEO**	Chemische Reaktionen	Chemische Reaktionen
	00:00-00:13	Einfache Umsetzung	Einfache Umsetzung
	00:13-00:31	Doppelte Umsetzung	Doppelte Umsetzung
VIDEO: Chemical Reactions: Synthesis and Decomposition	**GANZES VIDEO**	Chemische Reaktionen	Chemische Reaktionen
	00:00-00:14	Synthese	Synthese
	00:14-00:27	Stoffzerlegung	Stoffzerlegung
VIDEO: Chemische Stoffe: Amoniak Molekuel	**GANZES VIDEO**	Amoniak Molekuel	Chemische Stoffe
BILD: Faellungsreaktion Silbernitrat Natriumnitrat	**GANZES BILD** 	Chemische Reaktionen	Chemische Reaktionen
	Markierung 1	NaCl	Chemische Stoffe
	Markierung 2	AgNO3	Chemische Stoffe
	Markierung 3	AgCl	Chemische Stoffe
BILD: Quecksilberoxid Zerlegung	**GANZES BILD** 	Chemische Reaktionen, Stoffzerlegung	Stoffzerlegung
	Markierung 1	Quecksilber	Chemische Stoffe
	Markierung 2	Quecksilberoxid	Chemische Stoffe
	Markierung 3	Sauerstoff	Chemische Stoffe

Evaluation der Software für Multimedia Annotation (Variante B: ohne Prozessunterstützung)

Medien	Abschnitt	Schlagwörter	Einordnung: Thema
VIDEO: Classes of Chemical Reactions: Double Displacement Combustion	**GANZES VIDEO**	Chemische Reaktionen	Chemische Reaktionen
	00:00-00:15	Doppelte Umsetzung	Doppelte Umsetzung
	00:15-00:27	Verbrennung	Verbrennung
VIDEO: Classes of Chemical Reactions: Synthesis Decomposition	**GANZES VIDEO**	Chemische Reaktionen	Chemische Reaktionen
	00:00-00:13	Synthese	Synthese
	00:13-00:25	Stoffzerlegung	Stoffzerlegung
VIDEO: Chemische Stoffe: Stickstoff Molekuel	**GANZES VIDEO**	Stickstoff Molekuel	Chemische Stoffe
BILD: Stoffzerlegung von Quecksilberoxid 2	**GANZES BILD** 	Chemische Reaktionen, Stoffzerlegung	Stoffzerlegung
	Markierung 1	Quecksilber	Chemische Stoffe
	Markierung 2	Quecksilberoxid	Chemische Stoffe
	Markierung 3	Sauerstoff	Chemische Stoffe
BILD: Synthese von Kupfer Schwefel	**GANZES BILD** 	Chemische Reaktionen, Synthese	Synthese
	Markierung 1	Schwefelpulver	Chemische Stoffe
	Markierung 2	Kupferblech	Chemische Stoffe
	Markierung 3	Kupferschwefel	Chemische Stoffe

Evaluation der Software für Multimedia Annotation	Variante: mit Prozessunterstützung	Seite 1/3

Evaluation der Software für Multimedia Annotation

TECHNISCHE UNIVERSITÄT DARMSTADT

GRADUIERTENKOLLEG E-LEARNING

Fraunhofer IGD

Teilnehmernummer:______ Alter:______ Geschlecht:______

Im folgenden Fragebogen wirst Du zu Deiner Einschätzung der Software für Multimedia Annotation befragt. Bitte beantworte alle Fragen zügig und spontan, ohne lange über Deine Antwort nachzudenken.

Alle Angaben werden anonym behandelt und keinen weiteren Personen zugänglich gemacht.

Vielen Dank für Deine Mithilfe!

1. Allgemeine Einschätzung der Software

Die Software ist ...

schrecklich	☐ ☐ ☐ ☐ ☐	wunderbar
frustrierend	☐ ☐ ☐ ☐ ☐	zufriedenstellend
langweilig	☐ ☐ ☐ ☐ ☐	interessant
schwierig	☐ ☐ ☐ ☐ ☐	einfach
starr	☐ ☐ ☐ ☐ ☐	flexibel

2. Beurteilung der Benutzbarkeit der Annotationssoftware

Die Benutzungsoberfläche ...

	Stimme gar nicht zu				Stimme voll zu
ist einfach zu bedienen	☐	☐	☐	☐	☐
ist selbsterklärend bzgl. ihrer Funktionen und Bedienung	☐	☐	☐	☐	☐
ist einfach zu erlernen	☐	☐	☐	☐	☐
erfordert, dass man sich viele Details merken muss	☐	☐	☐	☐	☐
ist gut auf die Anforderungen der Aufgaben zugeschnitten	☐	☐	☐	☐	☐

Evaluation der Software für Multimedia Annotation	Variante: mit Prozessunterstützung	Seite 2/3

3. Beurteilung der allgemeinen Unterstützung von Arbeitsprozessen

Die Benutzungsoberfläche ...

	Stimme gar nicht zu				Stimme voll zu
bietet einen sehr guten Überblick über alle zu erledigenden Aufgaben	☐	☐	☐	☐	☐
bietet einen sehr guten Überblick über die Bearbeitungsabläufe	☐	☐	☐	☐	☐
zeigt deutlich auf, welche Aufgabe gerade erledigt werden muss	☐	☐	☐	☐	☐
macht deutlich sichtbar, welche Aufgaben bereits erledigt worden sind	☐	☐	☐	☐	☐
macht deutlich sichtbar, wie viele Medien im Prozessverlauf fertig bearbeitet werden	☐	☐	☐	☐	☐

4. Beurteilung des Konzepts zur Unterstützung von Arbeitsprozessen

Die Visualisierung der zu erledigenden Aufgaben ist ...

	Stimme gar nicht zu				Stimme voll zu
aufdringlich	☐	☐	☐	☐	☐
sehr gut steuerbar	☐	☐	☐	☐	☐
sehr hilfreich bei der Aufgabenerledigung	☐	☐	☐	☐	☐

Die Vorgabe der Aufgabenreihenfolge

(Medienauswahl→Kamerschnitte→Verschlagwortung→Einordnung in Struktur) ist ...

	Stimme gar nicht zu				Stimme voll zu
aufdringlich	☐	☐	☐	☐	☐
sehr hilfreich bei der Aufgabenerledigung	☐	☐	☐	☐	☐

Die Anzeige der aktuellen Aufgabe ist ...

	Stimme gar nicht zu				Stimme voll zu
sehr gut nachvollziehbar	☐	☐	☐	☐	☐
sehr hilfreich bei der Aufgabenerledigung	☐	☐	☐	☐	☐

Evaluation der Software für Multimedia Annotation	Variante: mit Prozessunterstützung	Seite 3/3

Die Anzeige der fertig bearbeiteten Aufgaben ist ...

	Stimme gar nicht zu				Stimme voll zu
sehr gut nachvollziehbar	☐	☐	☐	☐	☐
sehr hilfreich bei der Aufgabenerledigung	☐	☐	☐	☐	☐

Die Anzeige der fertig bearbeiteten Medien ist ...

	Stimme gar nicht zu				Stimme voll zu
sehr gut nachvollziehbar	☐	☐	☐	☐	☐
sehr hilfreich bei der Aufgabenerledigung	☐	☐	☐	☐	☐

Das automatische Ein-/ und Ausblenden von Werkzeugen ist ...

	Stimme gar nicht zu				Stimme voll zu
aufdringlich	☐	☐	☐	☐	☐
sehr hilfreich bei der Aufgabenerledigung	☐	☐	☐	☐	☐

Evaluation der Software für Multimedia Annotation	Variante: ohne Prozessunterstützung	Seite 1/2

Evaluation der Software für Multimedia Annotation

TECHNISCHE UNIVERSITÄT DARMSTADT

GRADUIERTENKOLLEG E-LEARNING

Fraunhofer IGD

Teilnehmernummer:______ Alter:______ Geschlecht:______

Im folgenden Fragebogen wirst Du zu Deiner Einschätzung der Software für Multimedia Annotation befragt. Bitte beantworte alle Fragen zügig und spontan, ohne lange über Deine Antwort nachzudenken.

Alle Angaben werden anonym behandelt und keinen weiteren Personen zugänglich gemacht.

Vielen Dank für Deine Mithilfe!

1. Allgemeine Einschätzung der Software

Die Software ist ...

schrecklich	☐☐☐☐☐	wunderbar
frustrierend	☐☐☐☐☐	zufriedenstellend
langweilig	☐☐☐☐☐	interessant
schwierig	☐☐☐☐☐	einfach
starr	☐☐☐☐☐	flexibel

2. Beurteilung der Benutzbarkeit der Annotationssoftware

Die Benutzungsoberfläche ...

	Stimme gar nicht zu				Stimme voll zu
ist einfach zu bedienen	☐	☐	☐	☐	☐
ist selbsterklärend bzgl. ihrer Funktionen und Bedienung	☐	☐	☐	☐	☐
ist einfach zu erlernen	☐	☐	☐	☐	☐
erfordert, dass man sich viele Details merken muss	☐	☐	☐	☐	☐
ist gut auf die Anforderungen der Aufgaben zugeschnitten	☐	☐	☐	☐	☐

Evaluation der Software für Multimedia Annotation	Variante: ohne Prozessunterstützung	Seite 2/2

3. Beurteilung der allgemeinen Unterstützung von Arbeitsprozessen

Die Benutzungsoberfläche …

	Stimme gar nicht zu				Stimme voll zu
bietet einen sehr guten Überblick über alle zu erledigenden Aufgaben	☐	☐	☐	☐	☐
bietet einen sehr guten Überblick über die Bearbeitungsabläufe	☐	☐	☐	☐	☐
zeigt deutlich auf, welche Aufgabe gerade erledigt werden muss	☐	☐	☐	☐	☐
macht deutlich sichtbar, welche Aufgaben bereits erledigt worden sind	☐	☐	☐	☐	☐
macht deutlich sichtbar, wie viele Medien im Prozessverlauf fertig bearbeitet werden	☐	☐	☐	☐	☐

Evaluation der Software für Multimedia Annotation	Abschliessende Befragung	Seite 1/2

Evaluation der Software für Multimedia Annotation

TECHNISCHE UNIVERSITÄT DARMSTADT

GRADUIERTENKOLLEG E-LEARNING

Fraunhofer IGD

Teilnehmernummer:______ Alter:______ Geschlecht:______

1. Allgemeine Fragen

Wie würdest Du Deine Kenntnisse im Umgang mit Computern einschätzen?

keine Kenntnisse ☐ ☐ ☐ ☐ ☐ sehr gute Kenntnisse

Wie würdest Du Deine Vorkenntnisse zu den Themen „Die Grundarten Chemischer Reaktionen" und „Chemische Stoffe" einschätzen?

keine Vorkenntnisse ☐ ☐ ☐ ☐ ☐ sehr gute Vorkenntnisse

Gab es generell Teilaufgaben, bei denen Du besondere Schwierigkeiten hattest? Wenn ja, beschreibe diese kurz.

2. Beide Programm-Varianten im Vergleich

Wie würdest Du generell urteilen: Welche der beiden Programm-Varianten ist besser für die Bearbeitung der Aufgaben geeignet?

ohne Prozess-Unterstützung ☐ ☐ mit Prozess-Unterstützung

Mit welcher der beiden Varianten hattest Du die meisten Schwierigkeiten?

ohne Prozess-Unterstützung ☐ ☐ mit Prozess-Unterstützung

Mit welcher der beiden Varianten kann man die Benutzung der Software besser erlernen?

ohne Prozess-Unterstützung ☐ ☐ mit Prozess-Unterstützung

Evaluation der Software für Multimedia Annotation	Abschliessende Befragung	Seite 2/2

Am neuen Konzept zur Unterstützung von Arbeitsprozessen finde ich gut:

____

____

____

____

Am neuen Konzept zur Unterstützung von Arbeitsprozessen finde ich nicht gut / könnte man besser machen:

____

____

____

____

Appendix D

User Study: Evaluation of Process-driven User Assistance.

D.1 Multiple Choice Items

	Frequency	Percent
1. The workflow visualization is not obstrusive.		
neutral	5	14.7%
agree	**19**	**55.9%**
strongly agree	10	29.4%
2. The workflow visualization is well controllable.		
strongly disagree	1	2.9%
neutral	4	11.8%
agree	**20**	**58,8%**
strongly agree	9	26.5%
3. The workflow visualization is helpful at task processing.		
strongly disagree	1	2.9%
agree	**18**	**52.9%**
strongly agree	15	44.1%

Table D.1: Ratings on the included workflow visualization panel (N = 34).

	Frequency	Percent
1. The predetermination of task order is not obstrusive.		
disagree	2	5.9%
neutral	3	8.8%
agree	**20**	**58.8%**
strongly agree	9	26.5%
2. The predetermination of tasks order is helpful at task processing.		
agree	15	44.1%
strongly agree	**19**	**55.9%**

Table D.2: Ratings on the predetermination of task order (N = 34).

	Frequency	Percent
1. The visualization of the recent task is well comprehensible.		
neutral	1	2.9%
agree	**19**	**55.9%**
strongly agree	14	41.2%
2. The visualization of the recent task is helpful at task processing.		
neutral	1	2.9%
agree	15	44.1%
strongly agree	**18**	**52.9%**

Table D.3: Ratings on the visualization of the recent task (N = 34).

	Frequency	Percent
1. The visualization of already accomplished tasks is well comprehensible.		
neutral	1	2.0%
agree	**20**	**58.8%**
strongly agree	13	38.2%
2. The visualization of already accomplished tasks is helpful at task processing.		
disagree	1	2.9%
neutral	4	11.8%
agree	**16**	**47.1%**
strongly agree	13	38.2%

Table D.4: Ratings on the visualization of accomplished tasks (N = 34).

	Frequency	Percent
1. The visualization of completed media documents is well comprehensible.		
neutral	5	14.7%
agree	**18**	**52.9%**
strongly agree	11	32.4%
2. The visualization of completed media documents is helpful at task processing.		
disagree	2	5.9%
neutral	6	17.6%
agree	**15**	**44.1%**
strongly agree	11	32.4%

Table D.5: Ratings on the visualization of completed media documents (N = 34).

	Frequency	Percent
1. The automatic invocation of tools is not obstrusive.		
strongly disagree	2	5.9%
disagree	2	5.9%
neutral	5	14.7%
agree	7	20.6%
strongly agree	**18**	**52.9%**
2. The automatic invocation of tools is helpful at task processing.		
strongly disagree	1	2.9%
disagree	1	2.9%
neutral	1	2.9%
agree	8	23.5%
strongly agree	**23**	**67.6%**

Table D.6: Ratings on the automatic supply of tools (N = 34).

D.2 Free-text Answers

	Positive	Negative
1	übersichtlicher, einfacher und schneller zu verstehen	evtl. etwas größere Schrift in der Leiste, welche die einzelnen Schritte anzeigt
2	dass die Fenster automatisch nach der Reihenfolge geöffnet werden. Man muss dadurch nicht mehr nachdenken, wann welcher Schritt gemacht werden muss	–
3	Automatische Einblendung der Arbeitsschritte. Einblendung der schon erledigten Aufgaben. Schnelle Bearbeitung der zu erledigenden Aufgaben. -Visualisierung	Aufgabenreihenfolge selbst bestimmen (erst Medien dann Bilder...)
4	Automatismen	Speicherung beim Schneiden vorab/während des Schneidens aktivierbar
5	Spart Zeit und Nachdenken	–
6	dass man durch die komplette Bearbeitung unterstüzt wird, somit entstehen keine Situationen bei welchen man nicht weiss, was zu tun ist. Die Bearbeitung geschieht schneller	Einstellung in Semantik:Es sollte eine farbliche Abgrenzung zwischen zB. chemischer Stoffe & chemische Reaktion stattfinden
7	dass man sieht, was man schon gemacht hat und bei welchem Bearbeitungsschritt man gerade ist	
8	bessere Orientierung dank Anzeige bereits erledigte Aufgaben noch ausstehende Aufgaben	bessere (größere Platzierung der Anzeige
9	Automatisches Einblenden der nächsten Arbeitsschrittes	Anzeige, wenn Tag vergessen wurde
10	Weniges Klicken, man wird geführt	–
11	dass es absolut weniger zeitaufwändig ist, einige klein Zwischenschritte für einen übernimmt	Durch das Übernehmen der Auswahl für die nächste Funktion wird einem dieser Schritt zwar abgenommen(gut, wenn man sich schon gut auskennt) aber demnach muss man auch nicht so viel mitdenken und denkt
12	das man mitverfolgen kann, wo man sich gerade befindet	–

13	Für den Einstieg ist die Prozessunterstützung sehr sinnvoll.Auch für Software-Laien wird die Benutzung somit zum Kinderspiel	keine Kritik allerdings wäre es gut, wenn man zwischen 'mit Prozessunterstützung' und 'ohne Prozessunterstützung' entscheiden könnte, damit der User flexible sein kann
14	dass man die Schritte gut folgen kann	–
15	Anzeige der Bearbeitungsstatus	Bevormundung durch die Software
16	–	–
17	Gute Strukturierung bzw. Überblick. Man gut sehen kann, bei welchem Schritt man gerade ist. Dass der Computer fast alles selbst macht	Geschwindigkeit
18	die Visualisierung der Arbeitsschritte/ wie weit man mit der Aufgabenbewältigung ist	–
19	dass man sich der Ablauf nicht genau merken muss, da er vorgegen ist. Die bessere Übersicht	–
20	Die Infos über die Workflowvisualisierung: wo bin ichm was muss noch erledigt werden. Man erlernt schneller & sinnvoller damit umzugehen	Tastatur als Bedienelement mit nutzen zu können. Nach Drag&Drop nochmal eine Anzeige/Info, das alles erfolgreich gespeichert wurde evtl. unnötig wenn Programm schneller läuft.
21	Es fällt leichter zu erfüllen, weil man keinen Schritt vergessen kann!	Das Video bei der Unterteilung aus Beginn jeweils nicht in Echtzeit, sondern schneller laufen lassen.
22	dass man weiss, wie viele Aufgaben man schon erledigt hat und wie viele man noch erledigen muss (Balken)	noch besser kenntlich machen, dass man zB. einen Tag gespeichert hat und dass man per 'Drag&Drop' die Kategorisierung vorgenommen hat (vielleicht so etwas wie (xy wurde zu 'Stoffzerlegung hinzugefügt')
23	dass die einzelnen Schritte bereits miteinander eingeblendet werden und man nicht selbst suchen muss, was der nächstes zu tun ist	Der Aufbau scheint anfangs verwirrend .. also etwas übersichtlicher gestalten vielleicht.
24	Die Unterstützung hinsichtlich der Visualisierung leichteres Merken von Abläufen	–

25	Dass das Programm den Nutzer durch die einzelnen ANwendungen fühlst und man so mit der Reihenfolge der einzelnen Teischritte nicht durcheinander kommt. Und dass man alle Dateien auf einmal laden kann	dass man nicht nur per Zahlenindex (zB. 3/5) sehen kann wieviel bereits bearbeitet wurde, sondern auch wie die bearbeiteten Dateien heißen, zB. V:1/5 chemische Reaktionen B:2/5 ... mit V fürVideo
26	–	–
27	Standpunkt der Arbeit immer ersichtlich. Keine Buttons suchen. Logische Anfolge. Man kann nichts vergessen	Allgemein: Schriftgröße erhöhen. Start-Button deutlicher kennzeichnen. Einzelne Teile des Videos immer aufgeklappt (Zieht sich nicht beim Tab-Speichern ein)
28	Besserer Überblick über Prozess. Überblick über bereits erledigte Aufgaben. -utomatisches Einblenden des Tools	–
29	schnell zu erlernen, übersichtlich, leitet einen durch das zu berarbeitenden Prozess	Geschwindigkeit bei der Einordnung in die Themenbereiche, Korrekturmöglichkeit während des Bearbeitung (zB. bei der Markierung von Bildern)
30	man findet sich schneller zurecht, Tool wird automatisch ausgewählt, man kann mehrere Videos und Bilder laden, man kann ablesen, wie viel man schon bearbeitet hat	unflexibel
31	guter Überblick, gute Visualisierung	–
32	den Überblick des Prozesses	den Prozess etwas größer (besser lesbar) darstellen
33	vorgegebener Weg, Übersicht über den Laufenden	die bereits abgeschlossenen Prozesse/ Arbeitsschritte & Gestaltung überarbeiten
34	dass man sieht welche Schritte man erledigt hat und noch erledigen muss. dass man bei Fehlern nochmal einen Schritt zurück gehen kann. dass die Symbole besser erklärt sind	–

Table D.7: Free-text Ratings on Process-driven User Assistance (N = 34).

List of Figures

List of Tables

Bibliography

[AAG06] Xavier Amatriain, Pau Arumi, and David Garcia. Clam: a framework for efficient and rapid development of cross-platform audio applications. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 951–954, New York, NY, USA, 2006. ACM.

[ABDF07] Maristella Agosti, Giorgetta Bonfiglio-Dosio, and Nicola Ferro. A historical and contemporary study on annotations to derive key features for systems design. *Int. J. Digit. Libr.*, 8(1):1–19, 2007.

[Ade97] Martin Ader. Seven workflow engines reviewed. *Document World*, 2(3):19–26, 1997.

[AF03] Maristella Agosti and Nicola Ferro. Annotations: Enriching a digital library. In Traugott Koch and Ingeborg Sølvberg, editors, *Research and Advanced Technology for Digital Libraries, 7th European Conference, ECDL 2003, Trondheim, Norway, August 17-22, 2003, Proceedings*, volume 2769 of *Lecture Notes in Computer Science*, pages 88–100. Springer, 2003.

[AF05a] Maristella Agosti and Nicola Ferro. Annotations as context for searching documents. In Fabio Crestani and Ian Ruthven, editors, *Context: Nature, Impact, and Role - 5th International Conference on Conceptions of Library and Information Sciences, CoLIS 2005, Glasgow, UK, June 4-8, 2005, Proceedings*, volume 3507 of *Lecture Notes in Computer Science*, pages 155–170. Springer, 2005.

[AF05b] Maristella Agosti and Nicola Ferro. A system architecture as a support to a flexible annotation service. In Can Türker, Maristella Agosti, and Hans-Jörg Schek, editors, *eer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures, 6th Thematic Workshop of the EU Network of Excellence DELOS, Cagliari, Italy, June 24-25, 2004, Revised Selected Papers*, volume 3664 of *Lecture Notes in Computer Science*, pages 147–166. Springer, 2005.

[AF06] Maristella Agosti and Nicola Ferro. Search strategies for finding annotations and annotated documents: The fast service. In Henrik Legind Larsen, Gabriella Pasi, Daniel Ortiz Arroyo, Troels Andreasen, and Henning Christiansen, editors, *Flexible Query Answering Systems, 7th International Conference, FQAS 2006, Milan, Italy, June 7-10, 2006, Proceedings*, volume 4027 of *Lecture Notes in Computer Science*, pages 270–281. Springer, 2006.

[AF08a] Maristella Agosti and Nicola Ferro. Annotations: A way to interoperability in DL. In Birte Christensen-Dalsgaard, Donatella Castelli, Bolette Ammitzbøll Jurik, and Joan Lippincott, editors, *Research and Advanced Technology for Digital Libraries, 12th European Conference, ECDL 2008, Aarhus, Denmark, September 14-19, 2008. Proceedings*, volume 5173 of *Lecture Notes in Computer Science*, pages 291–295. Springer, 2008.

[AF08b] Maristella Agosti and Nicola Ferro. A formal model of annotations of digital content. *ACM Trans. Inf. Syst.*, 26(1):3, 2008.

[AFFT04] Maristella Agosti, Nicola Ferro, Ingo Frommholz, and Ulrich Thiel. Annotations in digital libraries and collaboratories - facets, models and usage. In Rachel Heery and Liz Lyon, editors, *Research and Advanced Technology for Digital Libraries, 8th European Conference, ECDL 2004, Bath, UK, September 12-17, 2004, Proceedings*, volume 3232 of *Lecture Notes in Computer Science*, pages 244–255. Springer, 2004.

[AFPT06] Maristella Agosti, Nicola Ferro, Emanuele Panizzi, and Rosa Trinchese. Annotation as a support to user interaction for content enhancement in digital libraries. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 151–154, New York, NY, USA, 2006. ACM.

[AGH+98] Annette Adler, Anuj Gujar, Beverly L. Harrison, Kenton O'Hara, and Abigail Sellen. A diary study of work-related reading: design implications for digital reading devices. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 241–248, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[Ahe05] Terence C. Ahern. Using online annotation software to provide timely feedback in an introductory programming course. In *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference*, pages T2H–1, Oct. 2005.

[Als94] Robyn Alsop. Workflow automation integration requires a large technology toolkit and a structured approach. *Computer Technology Review*, page 2327, 1994.

[Anna] A.nnotate. `http://a.nnotate.com/`. [Online - accessed September 2010].

[Annb] Annotea Project. `http://www.w3.org/2001/Annotea/`. [Online - accessed September 2010].

[Anv] Anvil The video annotation research tool. `http://www.anvil-software.de/`. [Online - accessed September 2010].

[ARSF09] Marco Attene, Francesco Robbiano, Michela Spagnuolo, and Bianca Falcidieno. Characterization of 3d shape parts for semantic annotation. *Comput. Aided Des.*, 41(10):756–763, 2009.

[ASST05] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[Atl] ATLAS.ti the knowledge workbench. `http://www.atlasti.com/`. [Online - accessed September 2010].

[ATP+05] Thanos Athanasiadis, Vassilis Tzouvaras, Kosmas Petridis, Frederic Precioso, Yannis Avrithis, and Yiannis Kompatsiaris. Using a multimedia ontology infrastructure for semantic annotation of multimedia content. Proc. of 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot '05), Galway, Ireland, November 2005, 2005.

[AZ08] Omar Alonso and Hugo Zaragoza. Exploiting semantic annotations in information retrieval: Esair '08. *SIGIR Forum*, 42(1):55–58, 2008.

[BB07] Ralph Bobrik and Thomas Bauer. Towards configurable process visualizations with proviado. In *WETICE*, pages 367–369. IEEE Computer Society, 2007.

[BBGC01] A. J. Bernheim Brush, David Bargeron, Anoop Gupta, and J. J. Cadiz. Robust annotation positioning in digital documents. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 285–292, New York, NY, USA, 2001. ACM.

[BBN05] Marco Bertini, Alberto Del Bimbo, and Walter Nunziati. Automatic annotation of sport video content. In Alberto Sanfeliu and Manuel Lazo-Cortés, editors, *CIARP*, volume 3773 of *Lecture Notes in Computer Science*, pages 1066–1078. Springer, 2005.

[BCEB09] S. R. K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. *J. Artif. Int. Res.*, 34(1):569–603, 2009.

[BCL+04] Paolo Bottoni, Roberta Civica, Stefano Levialdi, Laura Orso, Emanuele Panizzi, and Rosa Trinchese. Madcow: a multimedia digital annotation system. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 55–62, New York, NY, USA, 2004. ACM.

[BCQ+04] Satanjeev Banerjee, Jason Cohen, Thomas Quisel, Arthur Chan, Yash Patodia, Ziad Al-Bawab, Rong Zhang, Alan Black, Richard Stern, Roni Rosenfeld, Alexander I. Rudnicky, Paul Rybski, and Manuela Veloso. Creating multi-modal, user-centric records of meetings with the carnegie mellon meeting recorder architecture. In *Proceedings of the ICASSP2004 Meeting Recognition Workshop, Montreal,Canada*, 2004.

[BD06] Jürgen Bortz and Nicola Döring. *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler.* Springer, Berlin, 4 edition, 2006.

[BDBCP04] Marco Bertini, Alberto Del Bimbo, Rita Cucchiara, and Andrea Prati. Semantic video adaptation based on automatic annotation of sport videos. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 291–298, New York, NY, USA, 2004. ACM.

[BDBT+07] Marco Bertini, Alberto Del Bimbo, Carlo Torniai, Costantino Grana, and Rita Cucchiara. Dynamic pictorial ontologies for video digital libraries annotation. In *MS '07: Workshop on multimedia information retrieval on The many faces of multimedia semantics*, pages 47–56, New York, NY, USA, 2007. ACM.

[BE10] Nizar Bouguila and Walid ElGuebaly. Integrating spatial and color information in images using a statistical framework. *Expert Syst. Appl.*, 37(2):1542–1549, 2010.

[BEP+08] Khalid Belhajjame, Suzanne M. Embury, Norman W. Paton, Robert Stevens, and Carole A. Goble. Automatic annotation of web services based on workflow definitions. *ACM Trans. Web*, 2(2):1–34, 2008.

[BFP09] Kerstin Bischoff, Claudiu S. Firan, and Raluca Paiu. Deriving music theme annotations from user tags. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 1193–1194, New York, NY, USA, 2009. ACM.

[BFar] Nadeem Bhatti and Dieter W. Fellner. *Modern Software Engineering Concepts and Practices: Advanced Approaches*, chapter Visual Semantic Analysis to support semi-automatic modelling of Services. (to appear).

[BFW07] R. M. Baecker, D. Fono, and P. Wolf. *Video Research in the Learning Sciences*, chapter Toward a Video Collaboratory Video research in the learning sciences, pages 461–478. Lawrence Erlbaum Associates, 2007.

[BGBG95] Ronald M. Baecker, Jonathan Grudin, William A.S. Buxton, and Saul Greenberg. *Readings in Human-Computer-Interaction: Toward the Year 2000*, chapter Cyberspace, pages 897–906. Morgan Kaufmann, San Francisco, CA, 1995.

[BHB+97] Sandford Bessler, Michael Hager, Hartmut Benz, Rolf Mecklenburg, and Steffen Fischer. Diane: A multimedia annotation system. In Serge Fdida and Michele Morganti, editors, *Multimedia Applications, Services and Techniques - ECMAST '97, Second European Conference, Milan, Italy, May 21-23, 1997, Proceedings*, volume 1242 of *Lecture Notes in Computer Science*, pages 183–198. Springer, 1997.

[Bib] Bibsonomy - a blue social bookmark and publication sharing system. `http://www.bibsonomy`. [Online - accessed September 2010].

[BKTT04] Peter Buneman, Sanjeev Khanna, Keishi Tajima, and Wang-Chiew Tan. Archiving scientific data. *ACM Trans. Database Syst.*, 29(1):2–42, 2004.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, pages 34–43, 2001.

[BLL+06] Paolo Bottoni, Stefano Levialdi, Anna Labella, Emanuele Panizzi, Rosa Trinchese, and Laura Gigli. Madcow: a visual interface for annotating web pages. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 314–317, New York, NY, USA, 2006. ACM.

[BLR03] Paolo Bottoni, Stefano Levialdi, and Paola Rizzo. An analysis and case study of digital annotation. In Nadia Bianchi-Berthouze, editor, *Databases in Networked Information Systems, Third International*

Workshop, DNIS 2003, Aizu, Japan, September 22-24, 2003, Proceedings, volume 2822 of *Lecture Notes in Computer Science*, pages 216–230. Springer, 2003.

[BMR+96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*. Wiley, 1 edition, 1996.

[BNPM93] Ronald M. Baecker, Dimitrios Nastos, Ilona R. Posner, and Kelly L. Mawby. The user-centered iterative design of collaborative writing software. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 399–405, New York, NY, USA, 1993. ACM.

[Boc03] Conrad Bock. Uml 2 activity and action models. *Journal of Object Technology*, 2(4):43–53, 2003.

[Bor77] Jürgen Bortz. *Statistik für Sozialwissenschaftler*. Springer, Berlin, 1977.

[BP05] Edwige Fangseu Badjio and Francois Poulet. User guidance: From theory to practice, the case of visual data mining. *IEEE International Conference on Tools with Artificial Intelligence*, 0:708–709, 2005.

[BPMI] Business Process Management Initiative. BPMN Graphical Elements. `http://www.bpmn.org/Samples/Elements/Core_BPMN_Elements.htm`. [Online - accessed September 2010].

[BPS+05] Stephan Bloehdorn, Kosmas Petridis, Carsten Saathoff, Nikos Simou, Vassilis Tzouvaras, Yannis Avrithis, Siegfried Handschuh, Yiannis Kompatsiaris, Steffen Staab, and Michael G. Strintzis. Semantic annotation of images and videos for multimedia analysis. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *The Semantic Web: Research and Applications: Proceedings of the Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29-June 1, 2005*, volume 3532 of *Lecture Notes in Computer Science*, pages 592–607. Springer, Berlin–Heidelberg, Germany, 2005.

[BR04] Hennie Brugman and Albert Russel. Annotating multi-media/multi-modal resources with elan. In *Proceedings of the 4th International Conference on Language Resources and Language Evaluation (LREC 2004)*, pages 2065–2068, Paris, 2004. European Language Resources Association.

[BR07] Matthias R. Brust and Steffen Rothkugel. On anomalies in annotation systems. In *Proceedings of the Third Advanced International Conference on Telecommunications (AICT 2007), May 13-19, 2007, Mauritius*, page 3. IEEE Computer Society, 2007.

[BRB05] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. Requirements for the visualization of system-spanning business processes. In *DEXA Workshops*, pages 948–954. IEEE Computer Society, 2005.

[BRB07] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. View-based process visualization. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *BPM*, volume 4714 of *Lecture Notes in Computer Science*, pages 88–95. Springer, 2007.

[Bru02] A.J. Bernheim Brush. Annotating digital documents: anchoring, educational use, and notification. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 542–543, New York, NY, USA, 2002. ACM.

[BTM02] Elisa Bertino, Alberto Trombetta, and Danilo Montesi. Workflow architecture for interactive video management systems. *Distrib. Parallel Databases*, 11(1):33–51, 2002.

[Bul03] Dick C. A. Bulterman. Using smil to encode interactive, peer-level multimedia annotations. In *DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering*, pages 32–41, New York, NY, USA, 2003. ACM.

[Cam] Camtasia Studio. `http://www.techsmith.de/camtasia.asp`. [Online - accessed September 2010].

[CB04] Jorge Cardoso and Robert P. Bostrom. Workflow management systems and erp systems: Differences, commonalities, and applications. *Inf. Technol. Manage*, 5:319–338, 2004.

[CBP99] Carlo Colombo, Alberto Del Bimbo, and Pietro Pala. Semantics in visual information retrieval. *IEEE MultiMedia*, 6:38–53, 1999.

[CCCJ08] Guillaume Cabanac, Max Chevalier, Claude Chrisment, and Christine Julien. Exploiting the annotation practice for personal and collective information management. In Sophie Ebersold, Agnès Front, Philippe Lopistéguy, Selmin Nurcan, Xavier Franch, Ela Hunt, and Remi Coletta, editors, *Proceedings of the International Workshop on Model Driven Information Systems Engineering: Enterprise, User and System Models (MoDISE-EUS'08) held in conjunction with the CAiSE'08 Conference, Montpellier, France, June 16-17, 2008*, volume 341 of *CEUR*

Workshop Proceedings, pages 67–78. CEUR-WS.org CEUR-WS.org, 2008.

[CCG02] Miguel Costa, Nuno Correia, and Nuno Guimar aes. Annotations as multiple perspectives of video content. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 283–286, New York, NY, USA, 2002. ACM.

[CCT+04] Dickson K. W. Chiu, S. C. Cheung, Sven Till, Kamalakar Karlapalem, Qing Li, and Eleanna Kafeza. Workflow view driven cross-organizational interoperability in a web service environment. *Inf. Technol. and Management*, 5(3-4):221–250, 2004.

[CDT06] Issam Chebbi, Schahram Dustdar, and Samir Tata. The view-based approach to dynamic inter-organizational workflow cooperation. *Data Knowl. Eng.*, 56(2):139–173, 2006.

[CDTT04] Panos Constantopoulos, Martin Doerr, Maria Theodoridou, and Manos Tzobanakis. On information organization in annotation systems. In *Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets*, pages 189–200, 2004.

[CFM+00] Bill Councill, Janet S. Flynt, Alok Mehta, John R. Speed, and Mary Shaw. Component-based software engineering and the issue of trust. In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 661–664, New York, NY, USA, 2000. ACM. Chairman-Heineman, George T.

[CG00] Donald Cox and Saul Greenberg. Supporting collaborative interpretation in distributed groupware. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 289–298, New York, NY, USA, 2000. ACM.

[CGG00] J. J. Cadiz, Anop Gupta, and Jonathan Grudin. Using web annotations for asynchronous collaboration around documents. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 309–318, New York, NY, USA, 2000. ACM.

[CH07] Robert Cannon and Fred Howell. Enhancing documents with annotations and machine-readable structured information using Notate. `http://www.textensor.com/enhancing-documents-2007.html`, March 2007. [Online; accessed 05-January-2010].

[CK91] Tzi-cker Chiueh and Randy Katz. An annotation system for vlsi design. In *Procedings of the International Conference on Computers and Communications*, 1991.

[Cla] CLAM Music Annotator. `http://clam-project.org/wiki/Music_Annotator`. [Online - accessed September 2010].

[CLM09] Marco Campanella, Riccardo Leonardi, and Pierangelo Migliorati. Interactive visualization of video content and associated description for semantic annotation. *Signal, Image and Video Processing*, 3(2):183–196, 2009.

[Cry] DART: Crystallography Annotation Tool. `http://www.itee.uq.edu.au/~eresearch/projects/dart/outcomes/crystallography.php`. [Online - accessed September 2010].

[CWX+07] Jingyu Cui, Fang Wen, Rong Xiao, Yuandong Tian, and Xiaoou Tang. Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 367–376, New York, NY, USA, 2007. ACM.

[CZF04] Teresa Chambel, Carmen Zahn, and Matthias Finke. Hypervideo design and support for contextualized learning. In Kinshuk, Chee-Kit Looi, Erkki Sutinen, Demetrios G. Sampson, Ignacio Aedo, Lorna Uden, and Esko Kähkönen, editors, *ICALT*. IEEE Computer Society, 2004.

[Db4] Db4o Oject Database. `http://www.db4o.com/`. [Online - accessed September 2010].

[Del] delicious social bookmarking. `http://del.icio.us`. [Online - accessed September 2010].

[Del05] Martin Delp. *A Reference Model for Media Production*. PhD thesis, Universität Stuttgart, 2005.

[DH95] James R. Davis and Daniel P. Huttenlocher. Shared annotation for cooperative learning. In *CSCL '95: The first international conference on Computer support for collaborative learning*, pages 84–88, Hillsdale, NJ, USA, 1995. L. Erlbaum Associates Inc.

[DNS03] Corsin Decurtins, Moira C. Norrie, and Beat Signer. Digital annotation of printed documents. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 552–555, New York, NY, USA, 2003. ACM.

[DOM] DOM4J. http://www.dom4j.org/. [Online - accessed September 2010].

[DS01] Jana Dittmann and Martin Steinebach. Platajanus: an audio annotation watermarking framework. In *MM&Sec '01: Proceedings of the 2001 Workshop on Multimedia and Security*, pages 44–47, New York, NY, USA, 2001. ACM.

[DSKT+09] Stijn De Saeger, Jun'ichi Kazama, Kentaro Torisawa, Masaki Murata, Ichiro Yamada, and Kow Kuroda. A web service for automatic word class acquisition. In *IUCS '09: Proceedings of the 3rd International Universal Communication Symposium*, pages 132–138, New York, NY, USA, 2009. ACM.

[DSP91] Glorianna Davenport, Thomas Aguirre Smith, and Natalio Pincever. Cinematic primitives for multimedia. *IEEE Comput. Graph. Appl.*, 11(4):67–74, 1991.

[DWC01] Thierry Declerck, Peter Wittenburg, and Hamish Cunningham. The automatic generation of formal annotations in a multimedia indexing and searching environment. In *Proceedings of the workshop on Human Language Technology and Knowledge Management*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[EAE+09] Mohamed Y. Eltabakh, Walid G. Aref, Ahmed K. Elmagarmid, Mourad Ouzzani, and Yasin N. Silva. Supporting annotations on relations. In *EDBT '09: Proceedings of the 12th International Conference on Extending Database Technology*, pages 379–390, New York, NY, USA, 2009. ACM.

[EBHR07] Samhaa R. El-Beltagy, Maryam Hazman, and Ahmed Rafea. Ontology based annotation of text segments. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 1362–1367, New York, NY, USA, 2007. ACM.

[EM97] Ernest A. Edmonds and Thomas P. Moran. Interactive systems for supporting the emergence of concepts and ideas. In *CHI '97: CHI '97 extended abstracts on Human factors in computing systems*, pages 233–233, New York, NY, USA, 1997. ACM.

[Exm] EXMARaLDA. http://www.exmaralda.org/. [Online - accessed September 2010].

[Far70] Mario V. Farina. *Flowcharting*. Prentice-Hall, 1970.

[Fau00] Gert Faustmann. Configuration for adaptation – a human-centeredapproach to flexible workflow enactment. *Comput. Supported Coop. Work*, 9(3-4):413–434, 2000.

[FB08] A. Felfernig and R. Burke. Constraint-based recommender systems: technologies and research issues. In *ICEC '08: Proceedings of the 10th international conference on Electronic commerce*, pages 1–10, New York, NY, USA, 2008. ACM.

[FBT+03] Ingo Frommholz, Holger Brocks, Ulrich Thiel, Erich Neuhold, Luigi Iannone, Giovanni Semeraro, Margherita Berardi, and Michelangelo Ceci. Document-centered collaboration for scholars in the humanities - the COLLATE system. In Sølvberg I.T. Koch, T., editor, *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003),*, pages 434–445, 2003.

[FFM04] Daniela Fogli, Giuseppe Fresta, and Piero Mussio. On electronic annotation and its implementation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 98–102, New York, NY, USA, 2004. ACM.

[FHM+01] Norbert Fuhr, Preben Hansen, Michael Mabe, András Micsik, and Ingeborg S. Digital libraries: A generic classification and evaluation scheme. In *ECDL '01: Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 187–199, London, UK, 2001. Springer-Verlag.

[Fin05] Matthias Finke. *Unterstützung des kooperativen Wissenserwerbs durch Hypervideo-Inhalte*. PhD thesis, Technische Universität Darmstadt, 2005.

[Flaa] Adobe Flash Player. `http://www.adobe.com/products/flex/`. [Online - accessed September 2010].

[Flab] Flash Player Penetration. `http://www.adobe.com/products/player_census/flashplayer/`. [Online - accessed September 2010].

[Fli] Flickr. `http://www.flickr.com/`. [Online - accessed September 2010].

[Fou] YAWL Foundation. YAWL: yet another workflow language. `http://www.yawlfoundation.org/`. [Online - accessed September 2010].

[FQA89] Richard Furuta, Vincent Quint, and Jacques Andre. Interactively editing structured documents. *Electron. Publ. Origin. Dissem. Des.*, 1(1):19–44, 1989.

[Fry94] Colleen Frye. Move to workflow provokes business process scrutiny. *Software Magazine*, 14(4):77–85, 1994.

[FTK04] Ingo Frommholz, Ulrich Thiel, and Thomas Kamps. Annotation-based document retrieval with four-valued probabilistic datalog. In Thomas Rölleke and Arjen P. de Vries, editors, *Proceedings of the first SIGIR Workshop on the Integration of Information Retrieval and Databases (WIRD'04)*, pages 31–38, Sheffield, UK, 2004.

[FX10] Songhe Feng and De Xu. Transductive multi-instance multi-label learning algorithm with application to automatic image annotation. *Expert Syst. Appl.*, 37(1):661–670, 2010.

[GCCG+04] Rudinei Goularte, Renan G. Cattelan, José A. Camacho-Guerrero, Valter R. Inácio, Jr., and Maria da Graça C. Pimentel. Interactive multimedia annotations: enriching and extending content. In *DocEng '04: Proceedings of the 2004 ACM symposium on Document engineering*, pages 84–86, New York, NY, USA, 2004. ACM.

[GGC+08] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. Video object annotation, navigation, and composition. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2008. ACM.

[GGR+05] Daniel Geraskov, Sven Göller, Wilfried Rüsse, Werner Sesink, and Thomas Trebing. Transformation einer vorlesung durch e-learning-elemente. *MedienPädagogik [Elektronische Ressource] : Online-Zeitschrift für Theorie und Praxis der Medienbildung ISSN 1424-3636*, 04-2, Januar 2005. http://www.medienpaed.com/04-2/sesink04-2.pdf.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[GHS95] Diimitrios Georgakopoulos, Mark Hornick, and Amit Sheth. An overview of workflow management: From process modeling to workflow automation infrastructure. In *DISTRIBUTED AND PARALLEL DATABASES*, pages 119–153, 1995.

[GKK+08] Spyridon V. Gogouvitis, George Kousiouris, Kleopatra Konstanteli, Theodoros Polychniatis, Andreas Menychtas, Dimosthenis Kyriazis, and Theodora Varvarigou. Realtime-enabled workflow management in service oriented infrastructures. In *AREA '08: Proceeding of the 1st ACM workshop on Analysis and retrieval of events/actions and workflows in video streams*, pages 119–124, New York, NY, USA, 2008. ACM.

[Goo] Google Earth. `http://earth.google.com/intl/de/`. [Online - accessed September 2010].

[GRS04] Birahim Gueye, Philippe Rigaux, and Nicolas Spyratos. Taxonomy-based annotation of xml documents: Application to elearning resources. In George A. Vouros and Themis Panayiotopoulos, editors, *Methods and Applications of Artificial Intelligence, Third Helenic Conference on AI, SETN 2004, Samos, Greece, May 5-8, 2004, Proceedings*, volume 3025 of *Lecture Notes in Computer Science*, pages 33–42. Springer, 2004.

[GVOH05] Joost Geurts, Jacco Van Ossenbruggen, and Lynda Hardman. Requirements for practical multimedia annotation. In *Workshop on Multimedia and the Semantic Web*, 2005.

[Hal88] Frank G. Halasz. Reflections on notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31:836–852, 1988.

[Han79] Patrick Hanks. *Collins Dictionary of the English Language*. William Collins, Glasgow, 1979.

[HB92] Beverly L. Harrison and Ronald M. Baecker. Designing video annotation and analysis systems. In *Proceedings of the conference on Graphics interface '92*, pages 157–166, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[HBF10] Cristian Hofmann, Uwe Boettcher, and Dieter W. Fellner. Change awareness for collaborative video annotation. In *Proceedings COOP 2010*. Springer, 2010. (to appear).

[HBR93] Lynda Hardman, Dick C.A. Bulterman, and Guido Rossum. The amsterdam hypermedia model: extending hypertext to support *real* multimedia. Technical report, Amsterdam, The Netherlands, The Netherlands, 1993.

[HBvR94] Lynda Hardman, Dick C. A. Bulterman, and Guido van Rossum. The amsterdam hypermedia model: adding time and context to the dexter model. *Commun. ACM*, 37(2):50–62, 1994.

[HF07] Sven Havemann and Dieter W. Fellner. Seven research challenges of generalized 3d documents. *IEEE Comput. Graph. Appl.*, 27(3):70–76, 2007.

[HF10a] Cristian Hofmann and Dieter Fellner. Supporting collaborative workflows of digital multimedia annotation. In *Proceedings of the 9th International Conference on Designing Cooperative Systems*. Springer, 2010. (to appear).

[HF10b] Cristian Hofmann and Dieter W. Fellner. Towards a workflow-based design of multimedia annotation systems. In *Proceedings of ED-MEDIA 2010 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2010. (to appear).

[HHD08] Nina Hollender, Cristian Hofmann, and Michael Deneke. Principles to reduce extraneous load in web-based generative learning settings. In *Workshop on Cognition and the Web 2008*, pages 7–14, 2008.

[HHF09a] Cristian Hofmann, Nina Hollender, and Dieter W. Fellner. Prozesse und abläufe beim kollaborativen wissenserwerb mittels computergestützter videoannotation. In Andreas Schwill and Nicolas Apostolopoulos, editors, *Lernen im digitalen Zeitalter - DeLFI 2009, die 7. E-Learning-Fachtagung Informatik der Gesellschaft für Informatik e.V., 14. - 17. September 2009 an der Freien Universität Berlin*, volume 153 of *LNI*, pages 115–126. GI, 2009.

[HHF09b] Cristian Hofmann, Nina Hollender, and Dieter W. Fellner. Task- and process-related design of video annotation systems. In Hartmut Wandke, Saskia Kain, and Doreen Struve, editors, *Mensch & Computer 2009: Grenzenlos frei!?, Interdisziplinäre Fachtagung, 6.-9. September 2009, Berlin, Germany*, pages 173–182. Oldenbourg Verlag, 2009.

[HHF09c] Cristian Hofmann, Nina Hollender, and Dieter W. Fellner. Workflow-based architecture for collaborative video annotation. In *Proceedings of the 13th International Conference on Human-Computer Interaction (HCII 09)*, pages 33–42. Springer, Springer, 2009.

[HHF09d] Cristian Hofmann, Nina Hollender, and Dieter W. Fellner. A workflow model for collaborative video annotation - supporting the workflow of collaborative video annotation and analysis performed in educational settings. In *Proceedings op the International Conference on Computer Supported Education (CSEDU 09)*, pages 199–204, Lissabon, Portugal, 22/03/2009 2009. INSTICC Press.

[HHK07] J. Hunter, M. Henderson, and I. Khan. Collaborative annotation of 3d crystallographic models. *J. Chem. Inf. Model*, 47(6):2475–2484, 2007.

[HHK08] Joey Hagedorn, Joshua Hailpern, and Karrie G. Karahalios. Vcode and vdata: illustrating a new framework for supporting the video annotation workflow. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 317–321, New York, NY, USA, 2008. ACM.

[Hil08] Dustin Lundring Hillard. *Automatic sentence structure annotation for spoken language processing*. PhD thesis, Seattle, WA, USA, 2008. Adviser-Ostendorf, Mari.

[HJK+09] Bernhard Haslhofer, Wolfgang Jochum, Ross King, Christian Sadilek, and Karin Schellner. The lemo annotation framework: weaving multimedia annotations with the web. *Int. J. Digit. Libr.*, 10(1):15–32, 2009.

[HM10] Guillaume Heusch and Sébastien Marcel. A novel statistical generative model dedicated to face recognition. *Image Vision Comput.*, 28(1):101–110, 2010.

[Hol95] David Hollingsworth. Workflow management coalition - the workflow reference model. Technical report, Workflow Management Coalition, January 1995.

[HR05] Arwen Hutt and Jenn Riley. Semantics and syntax of dublin core usage in open archives initiative data providers of cultural heritage materials. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 262–270, New York, NY, USA, 2005. ACM.

[HS94] Frank G. Halasz and Mayer Schwartz. The dexter hypertext reference model. *Communications of the ACM*, 37(2):30–39, February 1994.

[HS03] Siegfried Handschuh and Steffen Staab. *Annotation for the Semantic Web (Frontiers in Artificial Intelligence and Applications)*. IOS Press, January 2003.

[HSJ+09] Steffen Heinzl, Dominik Seiler, Ernst Juhnke, Thilo Stadelmann, Ralph Ewerth, Manfred Grauer, and Bernd Freisleben. A scalable service-oriented architecture for multimedia analysis, synthesis and consumption. *Int. J. Web Grid Serv.*, 5(3):219–260, 2009.

[HSM01] Siegfried Handschuh, Steffen Staab, and Alexander Maedche. CREAM — Creating relational metadata with a component-based, ontology driven framework. In *In Proceedings of K-Cap 2001*, Victoria, BC, Canada, October 2001.

[HWS07] Wu-Yuin Hwang, Chin-Yu Wang, and Mike Sharples. A study of multimedia annotation of web-based materials. *Comput. Educ.*, 48(4):680–699, 2007.

[IBM] IBMResearch. Efficient Video Annotation (EVA) system). http://domino.watson.ibm.com/comm/research.nsf/pages/r.multimedia.innovation.html. [Online - accessed September 2010].

[IMA+05] Yannis E. Ioannidis, David Maier, Serge Abiteboul, Peter Buneman, Susan B. Davidson, Edward A. Fox, Alon Y. Halevy, Craig A. Knoblock, Fausto Rabitti, Hans-Jörg Schek, and Gerhard Weikum. Digital library information-technology infrastructures. *Int. J. on Digital Libraries*, 5(4):266–274, 2005.

[ISO] ISO 9241-110:2006 Ergonomics of human-system interaction – Part 110: Dialogue principles. http://www.iso.org/iso/catalogue_detail.htm?csnumber=38009. [Online - accessed September 2010].

[JDC+95] David Jonassen, Mark Davidson, Mauri Collins, John Campbell, and Brenda Bannan Haag. Constructivism and computer-mediated communication in distance education. *American Journal of Distance Education*, 9(2):7–26, 1995.

[Jet] Jetty Webserver. http://jetty.codehaus.org/jetty/. [Online - accessed September 2010].

[Kan98] Joel P. Kanter. *Understanding thin-client/server computing*. Microsoft Press, Redmond, WA, USA, 1998.

[KBS10] Fanny Klett, Jan Bolz, and Christian Saul. D8.4.1 annual test report for ctc-wp5. *THESEUS Deliverables: WP8: Evaluation Task 8.7: Iterative System Design and Quality in Use*, 2010.

[KD09] Santanu Kolay and Ali Dasdan. The value of socially tagged urls for a search engine. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 1203–1204, New York, NY, USA, 2009. ACM.

[KGU10] Onur Küçüktunç, Uğur Güdükbay, and Özgür Ulusoy. Fuzzy color histogram-based video segmentation. *Comput. Vis. Image Underst.*, 114(1):125–134, 2010.

[Kha05] Cyrus D. Khazaeli. *Systemisches Design Intelligente Oberflächen für Information und Interaktion*. Rowohlt Verlag GmbH, 2005.

[Kie03] Bartosz Kiepuszewski. *Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows*. PhD thesis, Queensland University of Technology, Brisbane, Australia, 2003.

[Kip08] Michael Kipp. Spatiotemporal coding in anvil. In Bente Maegaard Joseph Mariani Jan Odjik Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 2042–2045, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[KK01] José Kahan and Marja-Ritta Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 623–632, New York, NY, USA, 2001. ACM.

[KLM+05] Rieko Kadobayashi, Julian Lombardi, Mark P. McCahill, Howard Stearns, Katsumi Tanaka, and Alan Kay. Annotation authoring in collaborative 3d virtual environments. In *ICAT '05: Proceedings of the 2005 international conference on Augmented tele-existence*, pages 255–256, New York, NY, USA, 2005. ACM.

[Kno] Knowtator. Managing annotation data. `http://knowtator.sourceforge.net/data-management.shtml`. [Online - accessed September 2010].

[LADM08] Ning Li, Abdelhak Attou, Suparna De, and Klaus Moessner. Device and service descriptions for ontology-based ubiquitous multimedia services. In *MoMM '08: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, pages 370–375, New York, NY, USA, 2008. ACM.

[LAH08] Massimiliano Leoni, W. M. Aalst, and A. H. Hofstede. Visual support for work assignment in process-aware information systems. In *BPM '08: Proceedings of the 6th International Conference on Business Process Management*, pages 67–83, Berlin, Heidelberg, 2008. Springer-Verlag.

[Las] Last.fm. `http://www.last.fm/`. [Online - accessed September 2010].

[LAT] Language Archiving Technology. ELAN. `http://www.lat-mpi.eu/tools/elan/`. [Online - accessed September 2010].

[LDK+08] Qing Luan, Steven M. Drucker, Johannes Kopf, Ying-Qing Xu, and Michael F. Cohen. Annotating gigapixel images. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 33–36, New York, NY, USA, 2008. ACM.

[Lin06] Daniel Link. *Computervermittelte Kommunikation im Spitzensport*. Sportverlag Strauß, 2006. Ph.D. Dissertation, Technische Universität Darmstadt.

[LK03] Ying Li and Chung-Chieh Kuo. *Video Content Analysis using Multimodal Information for Movie Content Extraction*. Kluwer Academic Publishers, 2003.

[LLL+10] Cheng-Yu Lu, Shian-Hua Lin, Jen-Chang Liu, Samuel Cruz-Lara, and Jen-Shin Hong. Automatic event-level textual emotion sensing using mutual action histogram between entities. *Expert Syst. Appl.*, 37(2):1643–1653, 2010.

[LSG+07] Michal Laclavik, Martin Seleng, Emil Gatial, Zoltan Balogh, and Ladislav Hluchy. Ontology based text annotation –ontea. In *Proceeding of the 2007 conference on Information Modelling and Knowledge Bases XVIII*, pages 311–315, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.

[Mad] MADCOW Multimedia Annotation of Digital Content Over the Web. `http://www.web-notes.com/index.php`. [Online - accessed September 2010].

[Mag83] Celeste S. Magers. An experimental evaluation of on-line help for non-programmers. In *CHI '83: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 277–281, New York, NY, USA, 1983. ACM.

[Mar97] Catherine C. Marshall. Annotation: from paper books to the digital library. In *DL '97: Proceedings of the second ACM international conference on Digital libraries*, pages 131–140, New York, NY, USA, 1997. ACM.

[Mar98] Catherine C. Marshall. Toward an ecology of hypertext annotation. In *HYPERTEXT '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 40–49, New York, NY, USA, 1998. ACM.

[Mat04] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. Computer Mediated Communication, LIS590CMC (Doctoral Seminar), 2004.

[May01] Richard E. Mayer. *Multimedia Learning*. Cambridge University Press, Cambridge, 2001.

[MB04] Catherine C. Marshall and A. J. Bernheim Brush. Exploring the relationship between personal and public annotations. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 349–357, New York, NY, USA, 2004. ACM.

[MBHM10] Christophe Millet, Isabelle Bloch, Patrick Hède, and Pierre-Alain Moëllic. Automatic cleaning and segmentation of web images based on colors to build learning databases. *Image Vision Comput.*, 28(3):317–328, 2010.

[MD89] Wendy E. Mackay and Glorianna Davenport. Virtual video editing in interactive multimedia applications. *Commun. ACM*, 32(7):802–810, 1989.

[MGF10] Iosif Mporas, Todor Ganchev, and Nikos Fakotakis. Speech segmentation using regression fusion of boundary predictions. *Comput. Speech Lang.*, 24(2):273–288, 2010.

[MJ06] M. Mikova and T. Janik. Analysis of elements supporting health in physical education: Methodological approach of video study. In Janík T. Wagner R. Muík, V., editor, *Muík, V., Janík, T., Wagner, R. (ed.) Neue Herausforderungen im Gesundheitsbereich an der Schule. Was kann der Sportunterricht dazu beitragen?*, pages 248–260, Brno, 2006. MU.

[MK10] John McDowall and Larry Kerschberg. A multi-agent approach for generating ontologies and composing services into executable workflows. In *EDBT '10: Proceedings of the 2010 EDBT/ICDT Workshops*, pages 1–12, New York, NY, USA, 2010. ACM.

[MKP02] José M. Martínez, Rob Koenen, and Fernando Pereira. Mpeg-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9:78–87, 2002.

[MLCG07] Nicolas Moreau, Michel Leclère, Michel Chein, and Alain Gutierrez. Formal and graphical annotations for digital objects. In *SADPI '07: Proceedings of the 2007 international workshop on Semantically aware document processing and indexing*, pages 69–78, New York, NY, USA, 2007. ACM.

[MMWF93] Raul Medina-Mora, Harry K. T. Wong, and Pablo Flores. Actionworkflow as the enterprise integration technology. *IEEE Data Eng. Bull.*, 16(2):49–52, 1993.

[MO10] Kevin McGuinness and Noel E. O'Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recogn.*, 43(2):434–444, 2010.

[Mon] M-Ontomat-Annotizer. `http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html`. [Online - accessed September 2010].

[Moo] Moodle. `http://moodle.org/`. [Online - accessed September 2010].

[MR02] Catherine C. Marshall and Christine Ruotolo. Reading-in-the-small: a study of reading on small form factor devices. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 56–64, New York, NY, USA, 2002. ACM.

[MR09] Tariq Mahmood and Francesco Ricci. Improving recommender systems with adaptive conversational strategies. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 73–82, New York, NY, USA, 2009. ACM.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[Nag03] Katashi Nagao. *Digital Content Annotation and Transcoding*. Artech House, Inc., Norwood, MA, USA, 2003.

[Nai96] Kshirasagar Naik. Automatic hardware synthesis of multimedia synchronizers from high-level specifications. *IEICE Transactions on Information and Systems*, E79-D(6):160–168, 1996.

[NBB+09] Kawa Nazemi, Matthias Breyer, Dirk Burkhardt, Christian Stab, Cristian Hofmann, and Christoph Hornung. D.ctc.5.7.1: Design und conceptualization semantic visualization framework. *THESEUS Deliverables: CTC-WP5 Innovative User Interfaces and Visualization*, 2, 2009.

[NCS09] Winyu Niranatlamphong, Worasit Choochaiwattana, and Michael B. Spring. A conceptual framework for digital annotation system on www. *Computer Science and Information Technology, International Conference on*, 0:27–31, 2009.

[NHK09] Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi. Canonical contextual distance for large-scale image annotation and retrieval. In

LS-MMRM '09: Proceedings of the First ACM workshop on Large-scale multimedia retrieval and mining, pages 3–10, New York, NY, USA, 2009. ACM.

[NKCM90] Christine M. Neuwirth, David S. Kaufer, Ravinder Chandhok, and James H. Morris. Issues in the design of computer support for co-authoring and commenting. In *CSCW '90: Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, pages 183–195, New York, NY, USA, 1990. ACM.

[NLW05] Eric W. T. Ngai, C. H. Leung, and Y. C. Wong. Application of the workflow management system in electronic commerce: a case study. *Int. J. Bus. Inf. Syst.*, 1(1/2):182–198, 2005.

[Nor93] Donald Norman. *Things that make us smart: defending human attributes in the age of the machine*. Perseus Books, Reading, 1993.

[NTTM09] Steven R. Ness, Anthony Theocharis, George Tzanetakis, and Luis Gustavo Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 705–708, New York, NY, USA, 2009. ACM.

[NUH09] Kawa Nazemi, Thomas Daniel Ullmann, and Christoph Hornung. Engineering user centered interaction systems for semantic visualizations. In *UAHCI '09: Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction. Addressing Diversity. Part I*, pages 126–134, Berlin, Heidelberg, 2009. Springer-Verlag.

[OAM99] Ilia A. Ovsiannikov, Michael A. Arbib, and Thomas H. McNeill. Annotation technology. *Int. J. Hum.-Comput. Stud.*, 50(4):329–362, 1999.

[Ogr06] Philip V. Ogren. Knowtator: a protégé plug-in for annotated corpus construction. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 273–275, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[OL10] Trevor M. O'Brien and David H. Laidlaw. Toward a machine learning approach for classifying user goals from user interactions. In *I3D '10: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 1–1, New York, NY, USA, 2010. ACM.

[OS97] Kenton O'Hara and Abigail Sellen. A comparison of reading paper and on-line documents. In *CHI '97: Proceedings of the SIGCHI conference* *on Human factors in computing systems*, pages 335–342, New York, NY, USA, 1997. ACM.

[Pai86] Allan Paivio. *Mental representations. A dual coding approach.* Oxford University Press, New York, 1986.

[Pan] Pandora radio. `http://www.pandora.com`. [Online - accessed September 2010].

[Pap09] Stavroula Papadopoulou. *MultiLevel Change Awareness for Collaborative Authoring Applications.* PhD thesis, ETH ZURICH, 2009.

[PAS+06] Kosmas Petridis, Dionysios Anastasopoulos, Carsten Saathoff, Norman Timmermann, Yiannis Kompatsiaris, and Steffen Staab. M-ontomat-annotizer: Image annotation linking ontologies and multimedia low-level features. In Bogdan Gabrys, Robert J. Howlett, and Lakhmi C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems, 10th International Conference, KES 2006, Bournemouth, UK, October 9-11, 2006, Proceedings, Part III*, volume 4253 of *Lecture Notes in Computer Science*, pages 633–640. Springer, 2006.

[Per02] Marcia Perry. Use scenarios for shared editing in scientific collaborations. In *Proceedings of the fourth International Workshop on Collaborative Editing, CSCW2002 Conference, November 16-20, 2002, New Orleans, Louisiana, United States.*, 2002.

[PH07] Roy Pea and Eric Hoffert. *Video Research in the Learning Sciences*, chapter Video workflow in the learning sciences: Prospects of emerging technologies for augmenting work practices, pages 427–460. Lawrence Erlbaum Associates. London, 2007.

[PJDM08] Gilberto Zonta Pastorello Jr., Jaudete Daltio, and Claudia Bauzer Medeiros. Multimedia semantic annotation propagation. In *ISM '08: Proceedings of the 2008 Tenth IEEE International Symposium on Multimedia*, pages 509–514, Washington, DC, USA, 2008. IEEE Computer Society.

[PLR06] Roy Pea, Robb Lindgren, and Joseph Rosen. Computer-supported collaborative video analysis. In *ICLS '06: Proceedings of the 7th international conference on Learning sciences*, pages 516–521. International Society of the Learning Sciences, 2006.

[PMR+04] Roy Pea, Michael Mills, Joseph Rosen, Kenneth Dauber, Wolfgang Effelsberg, and Eric Hoffert. The diver project: Interactive digital video repurposing. *IEEE MultiMedia*, 11(1):54–61, 2004.

[Pro] Probado: Prototypischer Betrieb allgemeiner Dokumente. `http://www.probado.de`. [Online - accessed September 2010].

[PW08] Artem Polyvyanyy and Mathias Weske. Flexible process graph: A prologue. In *OTM '08: Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part I on On the Move to Meaningful Internet Systems:*, pages 427–435, Berlin, Heidelberg, 2008. Springer-Verlag.

[QMXW10] Huimin Qian, Yaobin Mao, Wenbo Xiang, and Zhiquan Wang. Recognition of human activities using svm multi-class classifier. *Pattern Recogn. Lett.*, 31(2):100–111, 2010.

[Rat03] Donald Ratcliff. *Handbook of Qualitative Research in Psychology: Expanding Perspectives in Methodology and Design*, chapter Video Methods in Qualitative Research, pages 113–130. American Psychological Association, 2003.

[RBRB06] Stefanie Rinderle, Ralph Bobrik, Manfred Reichert, and Thomas Bauer. Business process visualization - use cases, challenges, solutions. In Yannis Manolopoulos, Joaquim Filipe, Panos Constantopoulos, and José Cordeiro, editors, *ICEIS 2006 - Proceedings of the Eighth International Conference on Enterprise Information Systems: Databases and Information Systems Integration, Paphos, Cyprus, May 23-27, 2006*, pages 204–211, 2006.

[RCES09] Abu Saleh Md. Mahfujur Rahman, Jongeun Cha, and Abdulmotaleb El Saddik. Authoring edutainment content through video annotations and 3d model augmentation. In *VECIMS'09: Proceedings of the 2009 IEEE international conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, pages 370–374, Piscataway, NJ, USA, 2009. IEEE Press.

[Rei06] Gerald Reif. Semantische annotation. *Semantic Web - Wege zur vernetzten Wissensgesellschaft*, 5(4):405–418, 2006.

[RFHB09] Kai Richter, Matthias Finke, Cristian Hofmann, and Dirk Balfanz. *Encyclopedia of Multimedia Technology and Networking*, chapter Hypervideo, pages 641–647. Idea Group Pub., USA, 2 edition, 2009.

[RH05] Lawrence Reeve and Hyoil Han. Survey of semantic annotation platforms. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1634–1638, New York, NY, USA, 2005. ACM.

[RH07] Lawrence H. Reeve and Hyoil Han. Conann: an online biomedical concept annotator. In *DILS'07: Proceedings of the 4th international conference on Data integration in the life sciences*, pages 264–279, Berlin, Heidelberg, 2007. Springer-Verlag.

[Rim05] Rolf Rimmele. Videograph. *IPN*, 2005.

[RK03] Sriram Ramachandran and Ramanujan Kashi. An architecture for ink annotations on web documents. In *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, page 256, Washington, DC, USA, 2003. IEEE Computer Society.

[Rob08] Charles Robert. Collaborative annotation creation and access in a multimodal environment with heterogeneous devices for decision support and for experience sharing. *Sensor Networks, Ubiquitous, and Trustworthy Computing, International Conference on*, sutc:519–523, 2008.

[SAYU01] Takeshi Sannomiya, Toshiyuki Amagasa, Masatoshi Yoshikawa, and Shunsuke Uemura. A framework for sharing personal annotations on web resources using xml. In *ITVE '01: Proceedings of the workshop on Information technology for virtual enterprises*, pages 40–48, Washington, DC, USA, 2001. IEEE Computer Society.

[SBCO01] Timothy J. Sliski, Matthew P. Billmers, Lori A. Clarke, and Leon J. Osterweil. An architecture for flexible, evolvable process-driven user-guidance environments. *SIGSOFT Softw. Eng. Notes*, 26(5):33–43, 2001.

[SBM08] Jürgen Steimle, Oliver Brdiczka, and Max Mühlhäuser. Coscribe: Using paper for collaborative annotations in lectures. In *ICALT*, pages 306–310. IEEE, 2008.

[SC94] John Sweller and Paul Chandler. Why some material is difficult to learn. *Cognition and Instruction*, 12(3):185–233, 1994.

[Sch92] August-Wilhelm Scheer. *Architektur Integrierter Informations-Systeme: Grundlagen Der Unternehmensmodellierung*. Springer, 1992.

[Sch00] Roland Schätzle. *Workflow-Management – ein ereignisbasierter Ansatz*. PhD thesis, PhD thesis at the Universität Karlsruhe (TH), Fakultät für Wirtschaftswissenschaften, 2000.

[Sch07] Rolf Schulmeister. *Grundlagen hypermedialer Lernsysteme. Theorie - Didaktik - Design.* Oldenbourg, München, 4 edition, 2007.

[Sch09] Thomas Schmidt. Creating and working with spoken language corpora in exmaralda. In *LULCL II: Lesser Used Languages & Computer Linguistics II*, pages 151–164, 2009.

[Sci91] Edward Sciore. Using annotations to support multiple kinds of versioning in an object-oriented database system. *ACM Trans. Database Syst.*, 16(3):417–438, 1991.

[SCJ+05] D. P. Spooner, J. Cao, S. A. Jarvis, L. He, and G. R. Nudd. Performance-aware workflow management for grid computing. *Comput. J.*, 48(3):347–357, 2005.

[SCL09] Eun-Joo Sin, Yoon-Chul Choy, and Soon-Bum Lim. Study on 3d annotation in virtual space for collaboration. In *CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering*, pages 382–387, Washington, DC, USA, 2009. IEEE Computer Society.

[Sem] SemaVis. `http://www.semavis.com`. [Online - accessed September 2010].

[SFZ06] Elmar Stahl, Matthias Finke, and Carmen Zahn. Knowledge acquisition by hypervideo design: An instructional program for university courses. *Journal of Educational Multimedia and Hypermedia*, 15(3):285–302, July 2006.

[SGF10] Federico M. Sukno, José J. Guerrero, and Alejandro F. Frangi. Projective active shape models for pose-variant image analysis of quasi-planar objects: Application to facial analysis. *Pattern Recogn.*, 43(3):835–849, 2010.

[SGL09] Hyunyoung Song, François Guimbretière, and Hod Lipson. The modelcraft framework: Capturing freehand annotations and edits to facilitate the 3d model design process using a digital pen. *ACM Trans. Comput.-Hum. Interact.*, 16(3):1–33, 2009.

[SGRH07] W. Sesink, S. Göller, G. Rößling, and D. Hofmann. emargo: Eine Digitale Randspalte zum Selbststudium (nicht nur) der Informatik. In *Proceedings der Pre-Conference Workshops der 5. e-Learning Fachtagung Informatik (DeLFI 2007)*, Berlin, 2007. Logos Verlag.

[Sha95] J. P. Shaffer. Multiple hypothesis testing. *Ann. Rev. Psych.*, 46:561584, 1995.

[Sha08] Robert M. Shapirod. Xpdl 2.1 integrating process interchange & bpmn. Technical report, Workflow Management Coalition, 2008.

[SHG+06] Ronald Schroeter, Jane Hunter, Jonathon Guerin, Imran Khan, and Michael Henderson. A synchronous multimedia annotation system for secure collaboratories. In *2nd IEEE International Conference on E-Science and Grid Computing (eScience*, page 41, 2006.

[SHJ+08] Dominik Seiler, Steffen Heinzl, Ernst Juhnke, Ralph Ewerth, Manfred Grauer, and Bernd Freisleben. Efficient data transmission in service workflows for distributed video content analysis. In *MoMM '08: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, pages 7–14, New York, NY, USA, 2008. ACM.

[Shn97] Ben Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 33–39, New York, NY, USA, 1997. ACM.

[Shn98] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, chapter Expert Reviews, Usablity Testing, surveys, and Continuing Asessments, pages 123–151. Pearson Addison Wesle, 3 edition, 1998.

[SHN07] Ronald Schroeter, Jane Hunter, and Andrew Newman. Annotating relationships between multiple mixed-media digital objects by extending annotea. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *The Semantic Web: Research and Applications, 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007, Proceedings*, volume 4519 of *Lecture Notes in Computer Science*, pages 533–548. Springer, 2007.

[Sik01] T. Sikora. The mpeg-7 visual standard for content description - an overview. 2001.

[SN92] Kevin J. Sullivan and David Notkin. Reconciling environment integration and software evolution. *ACM Trans. Softw. Eng. Methodol.*, 1(3):229–268, 1992.

[SOPS06] Giorgos Stamou, Jacco van Ossenbruggen, Jeff Z. Pan, and Guus Schreiber. Multimedia annotations on the semantic web. *IEEE MultiMedia*, 13(1):86–90, 2006.

[SPB05] Alexander Streit, Binh Pham, and Ross Brown. Visualization support for managing large business process specifications. In *Business Process Management*, pages 205–219, 2005.

[Spi04] J. A. Spiers. Tech tips: Using video management/ analysis technology in qualitative research. *International Journal of Qualitative Methods*, 3(1):57–61, 2004.

[SPK05] Tina Seidel, Manfred Prenzel, and Mareike (Eds.) Kobarg. *How to run a video study. Technical report of the IPN Video Study.* Waxmann, Münster, 2005.

[SPMG03] Frank M. Shipman, Morgan N. Price, Catherine C. Marshall, and Gene Golovchinsky. Identifying useful passages in documents based on annotation patterns. In Traugott Koch and Ingeborg Sølvberg, editors, *Research and Advanced Technology for Digital Libraries, 7th European Conference, ECDL 2003, Trondheim, Norway, August 17-22, 2003, Proceedings*, volume 2769 of *Lecture Notes in Computer Science*, pages 101–112. Springer, 2003.

[ST02] Janet M. Six and Ioannis G. Tollis. Automated visualization of process diagrams. In *GD '01: Revised Papers from the 9th International Symposium on Graph Drawing*, pages 45–59, London, UK, 2002. Springer-Verlag.

[Ste95] Ralf Steinmetz. *Multimedia-Technologie : Einführung und Grundlagen*. Springer, Berlin, 1995.

[SW05] Cees G. M. Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25:5–35, 2005.

[SW08] Han Sloetjes and Peter Wittenburg. Annotation by category: Elan and iso dcr. In Bente Maegaard Joseph Mariani Jan Odjik Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[Szy02] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing, Boston, MA, 2002.

[TAC+08] Christian Timmerer, Maria Teresa Andrade, Pedro Carvalho, Davide Rogai, and Giovanni Cordara. The semantics of mpeg-21 digital items revisited. In *MS '08: Proceeding of the 2nd ACM workshop on Multimedia semantics*, pages 17–23, New York, NY, USA, 2008. ACM.

[TBDW+05] Ulrich Thiel, Holger Brocks, Andrea Dirsch-Weigand, André Everts, Ingo Frommholz, and Adelheit Stein. Queries in context: Access to digitized historic documents in a collaboratory for the humanities. 2005.

[TG06] James Tam and Saul Greenberg. A framework for asynchronous change awareness in collaborative documents and workspaces. *Int. J. Hum.-Comput. Stud.*, 64(7):583–598, 2006.

[TH08] Chih-Fong Tsai and Chihli Hung. Automatically annotating images with keywords: A review of image annotation systems. *Recent Patents on Computer Science*, 1:55–68, 2008.

[The] THESEUS Neue Technologien für das Internet der Dienste. `http://www.theseus-programm.de`. [Online - accessed September 2010].

[THW+09] Jinhui Tang, Xian-Sheng Hua, Meng Wang, Zhiwei Gu, Guo-Jun Qi, and Xiuqing Wu. Correlative linear neighborhood propagation for video annotation. *Trans. Sys. Man Cyber. Part B*, 39(2):409–416, 2009.

[TO09] Katrin Tomanek and Fredrik Olsson. A web survey on the use of active learning to support annotation of text data. In *HLT '09: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 45–48, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[Toc95] Klaus Tochtermann. *Ein Modell für Hypermedia - Beschreibung und integrierte Formalisierung wesentlicher Hypermediakonzepte*. Shaker, Aachen, 1995. PhD thesis.

[Tog] Togehther XPDL Workflow Editor. `http://www.together.at/prod/workflow`. [Online - accessed September 2010].

[TTP07] Vassilis Tzouvaras, Raphaël Troncy, and Jeff Z. Pan. Multimedia annotation interoperability framework. Technical report, W3C Incubator Group Editor's Draft, 2007.

[Tur08] Douglas Ross Turnbull. *Design and development of a semantic music discovery engine*. PhD thesis, La Jolla, CA, USA, 2008. Adviser-Elkan, Charles and Adviser-Lanckriet, Gert.

[TW] Teddy Ward. Empiricism. `http://personal.ecu.edu/mccartyr/american/leap/empirici.htm`. [Online - accessed September 2010].

[UoQI] University of Queensland ITEE. Vannotea. `http://www.itee.uq.edu.au/~eresearch/projects/vannotea/index.html`. [Online - accessed September 2010].

[vdAH03] Wil M. P. van der Aalst and Arthur H. Hofstede. Yawl: Yet another workflow language. *Information Systems*, 30:245–275, 2003.

[vdAHKB03] Wil M. P. van der Aalst, Arthur H. Hofstede, Bartek Kiepuszewski, and Alistair P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51, 2003.

[vdAvH02] Wil M. P. van der Aalst and Kees van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.

[VSN05] Timo Volkmer, John R. Smith, and Apostol (Paul) Natsev. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 892–901, New York, NY, USA, 2005. ACM.

[VSS+07] Marko Vrhovnik, Holger Schwarz, Oliver Suhre, Bernhard Mitschang, Volker Markl, Albert Maier, and Tobias Kraft. An approach to optimize data processing in business processes. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 615–626. VLDB Endowment, 2007.

[W3Ca] W3C SOAP Specifications. `http://www.w3.org/TR/soap/`. [Online - accessed September 2010].

[W3Cb] W3C Web Services Architecture. `http://www.w3.org/TR/ws-arch/`. [Online - accessed September 2010].

[WBR+06] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. Elan: a professional framework for multimodality research. In *In Proceedings of Language Resources and Evaluation Conference (LREC*, 2006.

[WD02] Jörg Westbomke and Gisbert Dittrich. Towards an xml-based implementation of structured hypermedia documents. *Journal of Universal Computer Science*, 8:8–10, 2002.

[WDF+62] M. A. Witkin, R. B. Dyk, M. F. Faterson, D. R. Goodenough, and S. A. Karp. *Psychological differentiation*. Wiley, New York, 1962.

[WDH09] Jason Wither, Stephen DiVerdi, and Tobias Höllerer. Technical section: Annotation in outdoor augmented reality. *Comput. Graph.*, 33(6):679–689, 2009.

[WFM] Workflow Management Coalition. http://www.wfmc.org/. [Online - accessed September 2010].

[WFM99] Workflow Management Coalition Terminology & Glossary., February 1999.

[WHH+09] Meng Wang, Xian-Sheng Hua, Richang Hong, Jinhui Tang, Guo-Jun Qi, and Yan Song. Unified video annotation via multigraph learning. *IEEE Trans. Cir. and Sys. for Video Technol.*, 19(5):733–746, 2009.

[Whi03] Stephen A. White. Xpdl and bpmn. Technical report, Object Management Group/Business Process Management Initiative, 2003.

[Wit89] Merlin C. Wittrock. Generative processes of comprehension. *Educational Psychologist*, 24(4):345–376, 1989.

[WLD+10] Wei-Lin Wu, Ru-Zhan Lu, Jian-Yong Duan, Hui Liu, Feng Gao, and Yu-Quan Chen. Spoken language understanding using weakly supervised learning. *Comput. Speech Lang.*, 24(2):358–382, 2010.

[WMC] Workflow Management Coalition. XPDL public documents. http://www.wfmc.org/Specifications-Working-Documents/XPDL/View-category.html. [Online - accessed September 2010].

[WN01] Joanna L. Wolfe and Christine M. Neuwirth. From the margins to the center: The future of annotation. *Journal of Business and Technical Communication*, 15(3):333–371, 2001.

[WSM08] Christian Wolter, Andreas Schaad, and Christoph Meinel. Task-based entailment constraints for basic workflow patterns. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 51–60, New York, NY, USA, 2008. ACM.

[WWWCa] World Wide Web Consortium. Resource Description Framework (RDF). http://www.w3.org/RDF/. [Online - accessed September 2010].

[WWWCb] World Wide Web Consortium. Synchronized Multimedia. http://www.w3.org/AudioVideo/. [Online - accessed September 2010].

[WWWCc] World Wide Web Consortium. Web Ontology Language (OWL). http://www.w3.org/2004/OWL/. [Online - accessed September 2010].

[WZC08] Mei Wang, Xiangdong Zhou, and Tat-Seng Chua. Automatic image annotation via local multi-label classification. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 17–26, New York, NY, USA, 2008. ACM.

[WZY06] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM.

[XZW+09] Hongtao Xu, Xiangdong Zhou, Mei Wang, Yu Xiang, and Baile Shi. Exploring flickr's related tags for semantic annotation of web images. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM.

[YLS+04] Yun Yang, Wei Lai, Jun Shen, Xiaodi Huang, Jun Yan, and Lukman Setiawan. Effective visualisation of workflow enactment. In Jeffrey Xu Yu, Xuemin Lin, Hongjun Lu, and Yanchun Zhang, editors, *Advanced Web Technologies and Applications, 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, April 14-17, 2004, Proceedings*, volume 3007 of *Lecture Notes in Computer Science*, pages 794–803. Springer, 2004.

[yLTS03] Ching yung Lin, Belle L. Tseng, and John R. Smith. Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets. In *Proceedings of the TRECVID 2003 Workshop*, 2003.

[You] Youtube.com. `http://www.youtube.com`. [Online - accessed September 2010].

[ZB07] Daniel F. Zucker and Dick Bulterman. Open standard and open sourced smil for interactivity. *interactions*, 14(6):41–46, 2007.

[ZBM06] Qixing Zheng, Kellogg Booth, and Joanna McGrenere. Co-authoring with structured annotations. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 131–140, New York, NY, USA, 2006. ACM.

[ZBZ+08] Ding Zhou, Jiang Bian, Shuyi Zheng, Hongyuan Zha, and C. Lee Giles. Exploring social annotations for information retrieval. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 715–724, New York, NY, USA, 2008. ACM.

[ZHF+05] Carmen Zahn, Friedrich Hesse, Matthias Finke, Roy Pea, Michael Mills, and Joseph Rosen. Advanced digital video technologies to support collaborative learning in school education and beyond. In *CSCL*

'05: Proceedings of th 2005 conference on Computer support for collaborative learning, pages 737–742. International Society of the Learning Sciences, 2005.

[Zup06] Bernd Zupancic. *Vorlesungsaufzeichnungen und digitale Annotationen - Einsatz und Nutzen in der Lehre*. PhD thesis, Albert-Ludwigs Universität Freiburg, July 2006.

[ZZZ07] Ying Zou, Qi Zhang, and Xulin Zhao. Improving the usability of e-commerce applications using business processes. *IEEE Trans. Softw. Eng.*, 33(12):837–855, 2007.

Curriculum Vitae

	Cristian Erick Hofmann born on January 28, 1979 in Talca, Republica de Chile
1999	Abitur, Dietrich Bonhoeffer Gymnasium in Neunkirchen, Germany.
2000 - 2006	Studies of Computer Science with Applied Media Studies at the University of Siegen, Germany.
2003 - 2005	Student research assistant at the "Didactics of Informatics and eLearning" group at the University of Siegen, Germany.
March 2006	Diploma Thesis: "Entwurf und Implementierung einer Hypervideoanwendung für kooperativen Wissenserwerb" at the Computer Graphics Center (ZGDV) Darmstadt, Germany.
2006 - 2009	Scholarship holder of the DFG Postgraduate School "eLearning" at the Technische Universität Darmstadt, Germany.
January 2008	Two weeks research visit at Departamento de Informática, Universidade de Lisboa, Portugal.
since May 2009	Research associate at the Fraunhofer Institute for Computer Graphics Research Darmstadt, Germany. Projects: ◦ *THESEUS* (BMWi) WP5.5 Visual Semantics Annotation. ◦ *optimAAL* (BMBF) Kompetenzplattform für die Entwicklung und Einführung von AAL-Lösungen. ◦ *universAAL* (EU) Universal Open Platform and Reference Specification for Ambient Assisted Living.

Publications

Cristian Hofmann, Dirk Burkhardt, Matthias Breyer, Kawa Nazemi, Christian Stab, and Dieter W. Fellner: **Towards a Workflow-based Design of Multimedia Annotation Systems**. In *Proceedings of ED-MEDIA 2010 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*. AACE, 2010, pp.1224-1233.

Cristian Hofmann and Dieter W. Fellner: **Supporting Collaborative Workflows of Digital Multimedia Annotation**. In *Proceedings of the 9th International Conference on Designing Cooperative Systems*. Berlin, Heidelberg, New York: Springer, 2010, pp. 79-99.

Cristian Hofmann, Uwe Boettcher, and Dieter W. Fellner: **Change Awareness for Collaborative Video Annotation**. In *Proceedings of the 9th International Conference on Designing Cooperative Systems*. Berlin, Heidelberg, New York: Springer, 2010, pp. 101-117.

Christian Stab, Matthias Breyer, Kawa Nazemi, Cristian Hofmann, and Dieter W. Fellner: **SemaSun: Visualization of Semantic Knowledge based on an improved Sunburst Visualization Metaphor.** In *Proceedings of ED-MEDIA 2010 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*. AACE, 2010, pp. 911-919.

Dirk Burkhardt, Cristian Hofmann, Kawa Nazemi, Christian Stab, Matthias Breyer, and Dieter W. Fellner: **Intuitive Semantic-Editing for regarding Needs of Domain-Experts.** In *Proceedings of ED-MEDIA 2010 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*. AACE, 2010, pp. 860-869.

Nina Hollender, Cristian Hofmann, Michael Deneke, and Bernhard Schmitz: **Review: Integrating Cognitive Load Theory and Concepts of Human-Computer Interaction.** In *Comput. Hum. Behav.* 26(6), Nov. 2010, pp. 1278-1288.

Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **Workflow-Based Architecture for Collaborative Video Annotation**. In *Proceedings of the 3d international Conference on online Communities and Social Computing: Held As Part of HCI international 2009*. Berlin, Heidelberg, New York: Springer, 2009, Lecture Notes In Computer Science, vol. 5621, pp. 33-42.

Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **Task- and Process-related Design of Video Annotation Systems**. In *Proceedings of Mensch und Computer 2009: 9. fachübergreifende Konferenz für interaktive und kooperative Medien - Grenzenlos frei?!*. Berlin, Oldenbourg, 2009, pp. 173-182.

Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **Prozesse und Abläufe beim kollaborativen Wissenserwerb mittels computergestützter Videoannotation**. In *Proceedings of DeLFI 2009: 7. e-Learning Fachtagung Informatik der Gesellschaft für Informatik*. Berlin, 2009, Lecture Notes in Informatics, pp. 115-126.

Cristian Hofmann, Nina Hollender, and Dieter W. Fellner: **A WORKFLOW MODEL FOR COLLABORATIVE VIDEO ANNOTATION - Supporting the Workflow of Collaborative Video Annotation and Analysis performed in Educational Settings**. In *Proceedings of the International Conference on Computer Supported Education (CSEDU 09)*. INSTICC Press, 2009, pp. 199-204.

Kai Richter, Matthias Finke, Cristian Hofmann, and Dirk Balfanz: **Hypervideo**. In *Encyclopedia of Multimedia Technology and Networking*, 2nd ed., Idea Group Pub., USA, 2009, pp. 641-647.

Nina Hollender, Cristian Hofmann, Michael Deneke, and Bernhard Schmitz: **Cognitive load when exploring a video analysis database: Supporting comparisons and providing information on previous categorization processes.** *Poster presented at the 3rd International Cognitive Load Theory Conference, Heerlen, the Netherlands, 02/04/2009.*

Nina Hollender, Cristian Hofmann, and Michael Deneke: **Principles to reduce extraneous load in web-based generative learning settings**. In *Workshop on Cognition and the Web: Information Processing, Comprehension and Learning, Granada (Spain)*, 2008, pp. 7-14.

Cristian Hofmann, and Nina Hollender: **Kooperativer Informationsaustausch mit Hypervideo: Potentiale für das Web 2.0.** In *Proceedings der Pre-Conference Workshops der 5. e-Learning Fachtagung Informatik DeLFI 2007, Siegen*, 2007.

Cristian Hofmann, and Nina Hollender: **Informationserwerb und -Austausch durch Hypervideo.** In *Proceedings of Mensch Computer 2007: Konferenz für interaktive und kooperative Medien - Interaktion im Plural.* Berlin, Oldenbourg, 2007, pp. 269-272. (Awarded with the M&C-Praxispreis)

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind als soche kenntlich gemacht.

Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Cristian Erick Hofmann