The InViWo virtual agents

N. Richard (INRIA/ENST), P. Codognet (LIP6/INRIA) and A. Grumbach (ENST)

Nadine.Richard@inria.fr, Philippe.Codognet@lip6.fr, Alain.Grumbach@enst.fr

Abstract

The InViWo project aims at providing non-computer scientists with high-level tools to create inhabited virtual worlds, i.e. in which autonomous agents and avatars can interact. In this paper, we describe the agent model we designed for this project, and the integration of the concept of avatar in such a specific multi-agent system.

1. The InViWo project

1.1. Motivations

Many powerful 3D authoring tools already exist but they are generally designed for specialists like professional computer graphists (*3D Webmaster*, *3D Studio Max*, *VRCreator*). With such tools, one can create 3D shapes and animations, and sometimes program simple object behaviours.

There also are a few graphical tools for creating multi-user virtual worlds[1], or for describing character[2] or vehicle[3] behaviours using finite automatons.

Independently, there exists some tools for multi-agent system design, but they are usually built for computer scientists and often only used by their creators.

Eventually, there is no intuitive graphical tool to create both multi-user and multi-agent 3D worlds. Our goal is to design such a tool for people who do not necessarily have a deep knowledge in programming.

1.2. High-level tools

For the end-users, i.e. non-computer scientists, the InViWo (Intuitive Virtual Worlds) toolkit will include two different but complementary features:

- **A graphical user interface** (GUI) with a 3D view of the scene: the user will create the initial state of the world by selecting objects and relations (constraints, groups, *etc.*), and editing their properties and the associated behaviours. This GUI will provide high facilities for graphical programming (behaviours, interaction protocols, *etc.*) and component-based agent design.
- **A scripting language**: the advanced user will describe agent behaviours and relations in a declarative way. This language will essentially be based on primitives (i.e. the actions an agent can perform) and control structures that the user will combine to create more or less complex behaviours. The GUI will include a script editor.

1.3. Low-level library

The basic Java library is made of:

- An agent programming interface.
- An execution manager.

The GUI will use this API to concretely create the virtual world. Then the execution engine will animate the world.

Computer scientists and InViWo experts will be able to program pluggable modules for the API, in order to extend our agent model and develop new sensors, effectors and actions.

2. Virtual agents and avatars

We chose a simplified definition of the concept of agent: basically, an agent is an artificial entity that can perceive events and react to them depending on its own ressources and goals. Agent behaviours are sets of decision rules: the agent detects events with its sensors and performs the chosen actions *via* its effectors.

"Virtual agents" is buzzword for autonomous entities evolving in virtual worlds. Within traditional multi-user virtual worlds, the avatar (i.e. the user representation) is generally distinguished from the virtual agents, which are considered as "intelligent objects".

Published by the Eurographics Association, ISSN 1017-4656.

3. The InViWo agent model

We consider that each object is an agent because each object can possibly react to incoming events: an InViWo world is thus a multi-agent system.

3.1. The agent architecture

An InViWo agent can react to some stimuli coming from its environment (i.e. the other agents) by modifying its internal state (i.e. its properties) or by sending new stimuli into the environment: the stimulus is thus the basic element of inter-agent communication. Stimuli may also come from inside the agent, which then perceives its own state.

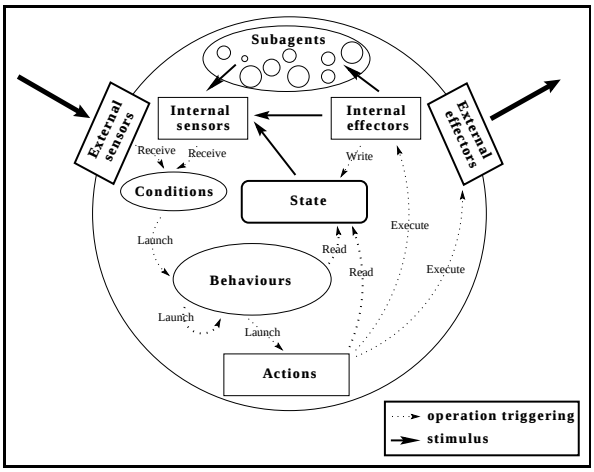

Figure 1: *Architecture of an InViWo agent.*

An agent may contain subagents; it is then considered as a multi-agent subsystem with its own characteristics. A subagent can be independent from its owner or under its control: for example, an active camera may be able to follow the agent or may be driven by the agent.

A condition is satisfied when all expected stimuli have been received. The condition then launches the corresponding behaviours. A behaviour can activate other behaviours or trigger appropriate actions. The set of actions an agent can perform represents its abilities.

The agent has external sensors and effectors, which correspond to in- and out-filters:

- A sensor can detect only some specific stimuli coming from the other agents.
- An effector can perform only some specified actions in a particular world.

Internal effectors can modify the internal state, which may send stimuli to inform that some values have changed. Internal sensors and effectors are also in charge of the communication between the agent and its subagents.

3.2. Constraints on agent behaviours and properties

Constraints are strong relationships between values; they should be kept satisfied by the associated solving mechanism. They are convenient tools for the description of particular behaviours. There are three different kinds of InViWo constraints, which are also described in [4] and [5]:

- **Internal constraints** are typical Constraint Programming[6] multi-way relations. They concern the variables of the internal state of the agent: it is a useful tool to control domain values.
- **External constraints** concern different agents: they can be considered as specific agents, which can give "orders" (or more exactly "hints") to the constrained agents when the constraint is violated.
- **Mixed constraints** (or functional constraints) are part of only one agent but use property values of other agents for their checks: if the constraint is getting unsatisfied, its owner should find a solution by itself.

Each agent contains its own constraint solver, and there is no need for a global solver to coordinate external constraints.

4. The InViWo avatars

An avatar is a specific agent which can be interactively controlled by a human operator. It can be a complex interface between a user and a virtual world: it both represents the user towards the other agents, and gives him a personal view of the environment and the capability to act on it.

From the human operator point of view, the avatar:

- controls the user interface in order to display information coming from the virtual world in a way the operator can understand it,
- and translates user actions into the multi-agent system communication mechanism (i.e. into existing stimuli).

As it is an agent, the avatar acts like a two-way filter (i.e. depending on its sensors and effectors) and may have its own behaviour. The user can thus partly control the avatar but he is constrained by its abilities and personality, and by the environment itself.

When autonomous enough, the avatar may assist the operator[7] by executing repetitive or high-precision tasks by itself.

5. Example

Let us illustrate our approach by an example which gathers the different types of agents we introduced. A woman (the user) is walking her artificial dog; the dog is kept on a leash. In such a case, we have three kinds of virtual agents:

- The woman is an avatar.
- The dog is a purely autonomous agent.
- The leash can be described as an external constraint.

The leash will be in charge of constraining the distance between the dog and its owner: it will send requests (e.g. next position advices) to both the woman and the dog when the constraint is violated. An "intelligent" leash will even be able to warn the associated agents before the constraint becomes unsatisfied.

Then, if the dog walks too quickly for instance, the user will have to decide between two reactions: hurry up or force the dog to slow down (with oral orders, or *via* the leash).

6. Conclusion

Our model tries to meet a multi-agent system approach with the needs in different types of inhabitants in virtual worlds (both autonomous and human-controlled entities). From the system point of view, each object or avatar is an agent with its own objectives, behaviours and abilities.

7. Availability

The agent model we proposed in this paper is now finalized, but the basic API of the InViWo toolkit is still under development. More information is available on the InViWo home page: `http://www.enst.fr/~richard/InViWo/`

References

1. *Cryo-Networks: the Scol language and technology*. `http://www.cryo-networks.com/`.
2. Y. Koga et al. *On intelligent digital actors.* In proc. of Imagina'98, France, 1998.
3. G. Moreau and S. Donikian. *From psychological and real-time interaction requirements to behavioural simulation.* In proc. Workshop on Computer Animation and Simulation'98, Portugal, September 1998.
4. N. Richard and P. Codognet. *Multi-way constraints for describing high-level objects behaviours in VRML*. In proc. of the AVI'98 Workshop on Intelligent Agents, Italy, May 1998.
5. N. Richard, P. Codognet and A. Grumbach. *Constraints to describe high-level behaviours in virtual worlds.* In proc. of the Eurographics'98 short-paper session, Portugal, September 1998.
6. V. Saraswat, P. Van Hentenryck, P. Codognet et al. *Constraint Programming.* ACM Computing Surveys, 28(4), December 1996.
7. D. Verna and A. Grumbach. *Sémantique et Localisation de l'Assistance en Réalité Virtuelle.* In proc. of GT-RV'98, France, March 1998.