Joint Virtual Reality Conference of EuroVR - EGVE (2011)
S. Coquillart, A. Steed, and G. Welch (Editors)

Integrating Semantic Directional Relationships into Virtual Environments: A Meta-modelling Approach

T.-H. Trinh, P. Chevaillier, M. Barange, J. Soler, P. De Loor, and R. Querrec

European University of Brittany, ENIB–UEB, LISYC, CERV, 29280 Plouzané, France

Abstract

This study is concerned with semantic modelling of virtual environments (VEs). A semantic model of a VE provides an abstract and high level representation of main aspects of the environment: ontological structures, behaviours and interactions of entities, etc. Furthermore, such a semantic model can be explored by artificial agents to exhibit human-like behaviours or to assist users in the VE. Previous research focused on formalising a knowledge layer that is a conceptual representation of scene content or application's entities. However, there still lacks of a semantic representation of spatial knowledge. This paper proposes to integrate a semantic model of directional knowledge into VEs. Such a directional model allows to specify relationships such as "left", "right", "above" or "north", "south" that are critical in many applications of VEs (e.g., VEs for training, navigation aid systems). We focus particularly on modelling, computing, and visualising directional relationships. First, we propose a theoretical model of direction in VEs that enables the specification of direction both from a first- and third-person perspective. Second, we propose a generic architecture for modelling direction in VEs using a meta-modelling approach. Directional relationships are described in a qualitative manner and at a conceptual level, and thus are abstract from metrical details of VEs. Finally, we show how our semantic model of direction can be used in a cultural heritage application to specify behaviours of artificial agents and to visualise directional constraints.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities, Evaluation/methodology

1. Introduction

Semantic modelling of virtual environments (VEs) provides an abstract, high-level and semantic description of main aspects of a VE: structure of the environment, behaviours/interactions of entities, and domain knowledge. An important motivation for such a semantic model is to facilitate the design of intelligent VEs hosting both natural and artificial agents (users and autonomous characters). Intelligence refers here to the ability of artificial agents to exhibit human-like behaviours, and to assist users to solve specific problems [AC01]. Among different approaches for semantic modelling of VEs, a common point is about the introduction into VEs a knowledge layer that is a conceptual representation of scene content [KCM06] or application's entities [LBW05].

Spatial knowledge is of central importance for a semantic representation of VEs [CTCC99]. It is referred not only as visual properties of entities but also semantic spatial relationships existing between them. Let us consider an example of a spatial expression that can be found in a VE: "*There is a red anchor disjoint with the ship. It is about 5 meters on the left of the ship.*". In this example, information such as color ("red", "blue"), shape ("round", "square"), or size ("large", "small") represents visual properties of entities. Meanwhile, topology ("disjoint", "overlap"), distance (in a quantitative - "5 meters" or qualitative - "near", "far" manner), or direction ("90 degrees" or "left", "right" or "north", "south") represent spatial relationships within a VE. Existing work (e.g., [BNC*03, KMDT*08]) attempted to semantically enrich VEs merely with visual properties of entities or multimedia landmarks such as texts, videos, images or Web links. However, spatial relationships are still classified as abstract information and thus, difficult to specify. To the best of our knowledge, there exists no previous work that has dealt with a semantic model for spatial relationships in VEs.

Nevertheless, in numerous cases, spatial relationships convey much more semantics about an environment than other spatial properties. Many researches on spatial cogni-

DOI: 10.2312/EGVE/JVRC11/067-074

tion [WLGB00], spatial navigation [DP02], and the transfer of spatial knowledge from virtual to real environments [WHK98] showed evidence of the role of spatial relationships in the human cognitive (or mental) representation of an environment. This provides a more flexible and abstract representation of the environment than other initial ones (e.g., landmark-based description). Moreover, some spatial properties are dependent on spatial relationships. For example, size of an entity depends on the distance from which the entity is observed. Similarly, different viewpoints (i.e., directions) can yield different shapes of the entity.

This paper introduces and integrates spatial relationships as a new kind of semantics into VEs. We tackle particularly directional relationships, an important family of spatial constraints. A semantic model of direction allows to describe relationships such as “in front”, ”above” or “south east”, “north” that are fundamental in many applications, like VEs for training, virtual tours, and navigation aid systems. In addition, as direction is used in everyday communication and human activities, an explicit representation of direction in VEs can be exploited by artificial agents to produce human-like behaviours or to assist users in many tasks such as searching an item or localising an object in the space. The main contributions of this paper are as follows:
- First, we propose a theoretical directional model. A difficulty is that direction is naturally strongly dependent on a reference system (e.g., an observer or magnetic poles). Different observers yield different views to a same directional relation. Our model allows to specify direction both from a first- and third-person perspective and thus provides a consistent representation of directional knowledge in the context of the collaboration between artificial agents and users. Furthermore, the proposed model serves as the basis for computing and visualising directional constraints in VEs.
- Second, we propose a generic architecture for modelling direction in VEs using a meta-modelling approach. A spatial language named VRX-OCL is proposed to formally specify direction at a conceptual level and in a qualitative manner. Direction modelling is abstract from metrical details of VEs.
- Finally, we apply our semantic directional model to produce intelligent behaviours of artificial agents in the context of a cultural heritage application. Human activities such as “*a worker must push a wheelbarrow in front of the ship*” can be specified and simulated. Moreover, the directional model is used to visualise directional areas (e.g., “*show the area on the left of the ship*”) and thus helps users in better localising objects or navigating within the environment.

In the next section, we briefly review related work. A detailed description of the directional model is given in Section 3. Section 4 describes the VRX-OCL spatial language and its architecture for modelling directions in VEs. Section 5 illustrates how our model has been applied to a cultural heritage application. We conclude the paper and outline some future work in Section 6.

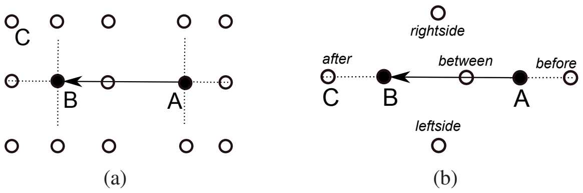
2. Related Work

2.1. Directional Models

Direction modelling has been studied in several areas, notably in artificial intelligence, geographical information systems, and spatial databases. Previous work converges on a qualitative representation of direction. That is, direction is described using qualitative prepositions such as “left of” instead of numerical terms such as “90 degrees”. A rationale for a qualitative approach is that it is often used in daily communication. It is considered to be closer to the human mental representation of space [CR08]. Currently, directional models differ from each others by the dimension of space (i.e., 2D or 3D space). This leads to a further distinction between exiting directional models based on how entities are represented in space. Most approaches merely consider 2D space, entities are simplified as points or regions. There exists very few models dealing with volumetric entities in 3D space.

The *double cross* model [Fre92] considers direction among three point-shaped objects in 2D, including a primary point and two reference points forming a vector. Based on the two lines orthogonal with the vector, 15 relative positions of the primary point to the vector can be distinguished (see Fig. 1(a)). [PET02] introduced an extension of the model to distinguish 75 relations among three arbitrary points in 3D. A more coarse model, called *projective model* [CB06], is based on the collinearity to define 5 relations among three points (see Fig. 1(b)). The projective model thereafter was extended to regions and volumetric objects in 3D space [BC06]. Although these models do not need an extrinsic reference system, they are not often used because three objects are required to formalise a direction.

Alternatively, the cardinal direction model [Fra92] defines direction between a primary point and a reference point in 3D. The model assumes the existence of a reference system such as the intrinsic orientation of the reference object (e.g., “front”, “back”) or magnetic poles (e.g., “north”, “south”). Such a reference system divides the space around the reference point into 8 cone-shaped areas (see Fig. 2(a)) or 4 partitions using a projection-based approach (see Fig. 2(b)). Moreover, a neutral zone (i.e., an area around the reference



(a) (b)

Figure 1: *Direction among three points (a point C and a vector $\vec{AB}$) in 2D space: (a) 15 relations are defined in the double cross model [Fre92]; (b) 5 relations are defined in the projective model [CB06].*

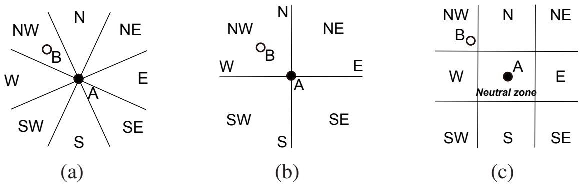


Figure 2: *Cardinal direction between two points in 2D space [Fra92]: (a) cone-based model; (b) projection-based model; (c) projection-based model with neutral zone.*

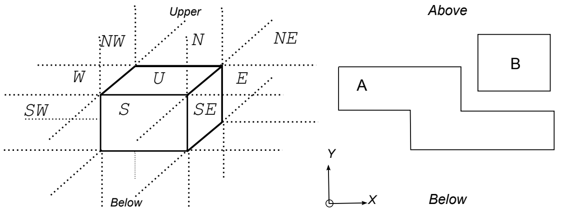


Figure 3: *Left: The TCD model [CLJZ07] defines 27 relations around a reference object approximated by its axis-aligned bounding box. Right: The model in [BR09] deals with direction between complex objects in 3D by projecting them into 2D planes*

object where no direction is defined) can be added to divide the 2D space into 9 areas as in Fig. 2(c).

[CLJZ07] extended the cardinal direction model to 3D space, called TCD (Three-dimensional Cardinal Direction) model. The reference object is approximated by its axis-aligned minimum bounding box that partitions the 3D space around it into 27 directional relations according to 3 layers (upper, medium, and below) (see Fig. 3 left). Because the TCD model did not take into account shapes of objects (concave and convex objects are treated in a same way using their bounding boxes), this may lead to wrong results in some situations. Some approaches tried to model directional relations among complex objects (such as stairs) by projecting them into 2D planes [BR09], as in Fig. 3 right, or by calculating the intersection between cubic matrix [CS10]. These approaches have big computational issues because they require to partition objects and the 3D space into cubic cells.

To summarize, the complexity of directional models is directly proportional with the multi-dimensionality of space. No directional model developed so far has efficiently dealt with complex objects in 3D. Approximation structures such as bounding boxes or points are often used to simplify the specification of direction between objects. Moreover, no current directional model has been proposed in the context of frames of reference, as presented in the following.

2.2. Frames of Reference

Most spatial relations must be given with respect to a *frame of reference* (FoR). For example, in distance models, the FoR can be the size of an object (e.g. a bus stop). In the case of direction, the FoR is a given direction from which a directional relation is observed. It can be an absolute direction (e.g., magnetic poles), or a relative direction (e.g., intrinsic orientation of a given object or a viewer). Three types of FoR can be distinguished: *(i)* intrinsic – direction is given by inner properties of the reference object (its intrinsic direction); *(ii)* extrinsic – direction is imposed by external factors on the reference object (e.g., a backward motion of the reference object changes its intrinsic direction); and *(iii)* deictic – direction is based on a third-person view from which the reference object is seen [Her94].

In VEs, [STV09] showed that users prefer a third-person view in some tasks such as navigation, meanwhile a first-person view is better in other tasks, such as manipulating an object with the hands. A switching between two views is also needed. Alternatively, [SB04] studied how different FoR could be combined in collaborative VEs. Users are embedded into VEs with two different roles (i.e., director and actor) to perform tasks. A director can provide instructions based on his perspective (e.g., "in front of me" or "go to my left") or the actor's perspective (e.g., "beside you"). The results showed that a combination of different FoR yields a better collaboration in some tasks.

Although the benefits of FoR in VEs have been recognised, no previous work in the field of VEs has dealt with the modelling of such an important concept. In the following section, we propose an integrated model of direction in VEs that takes into account direction both from a first- and third-person perspective with a clear definition of FoR.

3. An Integrated Directional Model for VEs

3.1. Modelling Directional Entities

To conceptualise direction in VEs, we propose the concept of *directional entity*. A directional entity encompasses a scene entity but with additional semantics. In VEs, a directional entity can be a spatial object (tangible or not), an artificial agent, or a user. It shares some properties with the model defined by [IDM06]. Every entity is uniquely identified in a VE by a *name*. An entity is graphically represented by one or more *shape(s)* using 3D formats such as VRML or X3D. Every entity has a *position* that enables to localise it in VEs. Usually, position of an entity is defined as its centroid and represented by a vector. Moreover, as discussed above, directional models dealing with 3D complex entities raise many representational and computational issues. In our model, we thus propose to simplify an entity by two levels of approximation. In the first case, an entity can be simplified to a *referential point*. A referential point represents some semantics about an entity, e.g., an interaction point or a navigation point. In the second case, entity is described by its *bounding box*. A bounding box allows a volumetric representation of an entity. Furthermore, a bounding box also conveys other important information about an entity, such as *width*, *height*, and *depth*. Such information eases the definition of directional relations between entities. An example of directional entity (e.g., a ship) is given in Fig. 5.

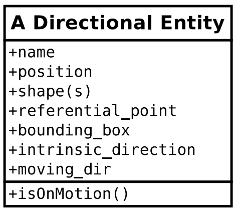


Figure 4: *The conceptual definition (in the form of a UML class) of directional entity in the proposed directional model.*

With regard to directional information, we propose that an entity is possibly oriented. That is, an entity can have an *intrinsic direction*. We represent such intrinsic direction by means of three unit vectors $\overrightarrow{front}$, $\overrightarrow{left}$, $\overrightarrow{above}$. Other intrinsic directions of an entity (e.g., "right", "back", and "below") can be inferred from these unit vectors. However, it is important to note that, in our model, an intrinsic direction is not an obligation for every entity in a VE. We are aware that one can easily identify the intrinsic direction of a ship, a house, etc. but not other objects like a sphere or a table. As we stated earlier, directional relations must be given with respect to a reference system. When an absolute reference system is used (e.g., magnetic poles), no intrinsic direction is required. It is similar when direction is defined from a third-person perspective.

Moreover, we have seen previously that external factors can have impacts on the description of direction. We are particularly interested in motion of entities. In our model, directional entities are not still but possibly moving (e.g., a ship can move in the sea; a virtual human can walk from an initial place to a target; a user is free to move in a VE). A motion is modelled by the *moving_dir* attribute that defines the front, left, and above direction of the motion.

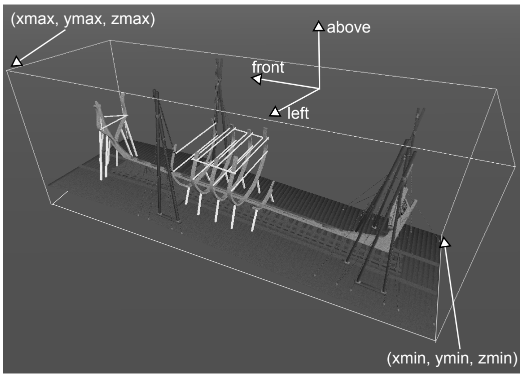


Figure 5: *Example of a directional entity in VEs.*

3.2. Modelling Frames of Reference

As discussed in Section 2.2, direction must be given within the context of an FoR that can be intrinsic, extrinsic, or deictic. However, in the context of VEs, direction is sometimes defined on a reference object without any intrinsic direction (e.g., a sphere or a table) and without mentioning any explicit FoR. An example could be "*In the room, the round table is on the left of the square table*". We thus propose that a VE has an implicit FoR that can be a directional entity assumed as a default viewer, or an absolute direction like magnetic poles. Such an implicit FoR is often used in (but not limited to) indoor spaces. For example, within a room, it is commonly accepted that the entrance door is the point of view by default to the room. In this case, the entrance door plays the role of the implicit FoR.

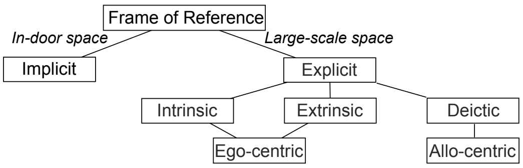


Figure 6: *Taxonomy of different FoR in VEs.*

Fig. 6 illustrates different FoR used in our model of direction in VEs. Using the concept of directional entity presented above, the use of each type of FoR is as follows.
- When direction is given with respect to a viewer (e.g., "*A sees that B is on the left of C*"), this is the case of deictic FoR. The modelling of such a viewer-based direction (also called allo-/exo-centric direction) is described in Sect. 3.4.
- Otherwise, when no viewer is given (e.g., "*B is on the left of C*"), direction is based on the reference object (entity C in the example), so called first-person perspective or egocentric direction (see Sect. 3.3). There are three possibilities. First, if the reference object C is moving (i.e., it is impacted by external factors), the direction of motion (defined by the *moving_dir* attribute) will be used as a reference system to compute directional relations. This is the case of extrinsic FoR. Second, if the reference object C is not moving and has an intrinsic direction, this intrinsic direction will be used. This is the case of intrinsic FoR. Finally, if the reference object C is not moving and has not an intrinsic direction, the implicit FoR defined in the VE will be used.

Obviously, such a mechanism allows a unambiguous modelling of FoR in our directional model.

3.3. Direction From a First-person Perspective

Direction based on a first-person perspective is also called as object-based or egocentric direction. Direction is dependent on the direction of motion and intrinsic direction of the reference object. Not lost generality, we assume that the reference object is still. Thus its intrinsic direction will be used.

Fig. 5 illustrates a ship (a directional entity) considered as the reference object. To model directional relations, we were inspired by the TCD model. That is, the reference object is approximated by its minimum bounding box. 27 directional relations divided into 3 layers (upper, medium, and below) can be defined. For each layer, 9 possible relations are "front", "front left", "front right", "left", "right", "behind", "behind left", "behind right", and "neutral". These relations have equivalent terms in geographical direction: "north", "north west", "north east", etc. Given (x_{min}, y_{min}, z_{min}) and (x_{max}, y_{max}, z_{max}) as two extreme points of the minimum bounding box, a formal definition of the *left* relation between a point (x, y, z) and the reference object is:

$x_{min} < x < x_{max}$ and $y > y_{max}$ and $z_{min} < x < z_{max}$

Interestingly enough, such a formalisation could be extended to take into account information on distance. For example, a point (x, y, z) is *on the left* and at *a distance d* from the ship is formalised as follows:

$x_{min} < x < x_{max}$ and $y_{max} < y < y_{max} + d$ and $z_{min} < x < z_{max}$

Other directional relations can be similarly formalised. These formalisations serve as the basis for visualising spatial constraints in VEs (see Sect. 5.2 and Fig. 12).

3.4. Direction From a Third-person Perspective

In this case, direction is observed from the perspective of a third-person. It is also referred to as viewer-based or allo-/exo-centric direction. Fig. 7 shows an example of direction from a third-person perspective, that is "*The worker sees that the anchor is on the left of the ship*". It is a simplified illustration in 2D of the scene in Fig. 11.

The calculation of direction is based on the intrinsic direction of the viewer, whereas both primary and reference objects may not be oriented. We are based on a vector-based directional algebra proposed in [SLC99]. Given three points W, A, and S respectively as referential points of the worker, the anchor, and the ship, the algebra computes the direction by the dot-product between the vector $\overrightarrow{SA}$ and the three directions (i.e., $\overrightarrow{front}$, $\overrightarrow{left}$, $\overrightarrow{above}$) of the viewer. Considering Fig. 7, it is easy to calculate that $\overrightarrow{SA} \odot W.\overrightarrow{front} = 0$ and $\overrightarrow{SA} \odot W.\overrightarrow{left} > 0$ and $\overrightarrow{SA} \odot W.\overrightarrow{above} = 0$. As a result, according to the vector-based directional algebra, it is concluded that A(nchor) is on the left of S(hip). Similarly, the conditions for A to be in front of S are: $\overrightarrow{SA} \odot W.\overrightarrow{front} > 0$ and $\overrightarrow{SA} \odot W.\overrightarrow{left} = 0$ and $\overrightarrow{SA} \odot W.\overrightarrow{above} = 0$. A combination of the two previous conditions allows to verify if A is in front and left of S: $\overrightarrow{SA} \odot W.\overrightarrow{front} > 0$ and $\overrightarrow{SA} \odot W.\overrightarrow{left} > 0$ and $\overrightarrow{SA} \odot W.\overrightarrow{above} = 0$. Other relations can also be computed.

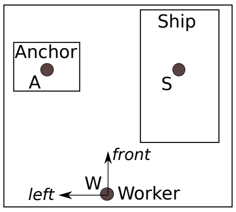


Figure 7: *Example of direction from a third-person view: "the worker sees that the anchor is on the left of the ship".*

4. The VRX-OCL Spatial Language

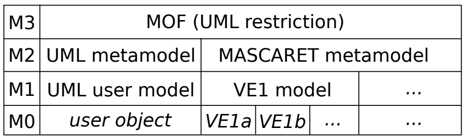
Based on our theoretical model of direction, this section discusses how to specify directional relations in VEs. The main goal is to enable the specification of directional constraints at a high level and in a formal way. To do so, we first propose a meta-modelling approach for conceptualising VEs. We then propose a spatial language named VRX-OCL to maintain constraints at the conceptual model of VEs.

4.1. Meta-modelling of Virtual Environments

To conceptualise VEs, we use MASCARET, a framework for the semantic modelling of VEs [CTB*11]. Using MASCARET, the design of VEs is based on a multi-layer architecture according to MOF proposal (MetaObject Facility - www.uml.org/mof) (see Fig. 8). MASCARET uses the Unified Modeling Language (UML) as a formal basis to semantically specify main aspects of a VE: the structure of environment; behaviours of entities; activities of users and agents. For the sake of clarity, we briefly present in the following respectively the M1, M0, and M2 layer in MASCARET, which have been presented in more details in [CTB*11].

The M1 layer corresponds to the conceptual model of a VE. In MASCARET, different UML diagrams are used to conceptualise a VE. The structure of a VE is represented by UML classes diagrams. For example, Fig. 9 illustrates the representation of ships and anchors. Conceptual relations between domain concepts are represented by UML associations. Instances of a class are directional entities.

Once the conceptual model of a VE is defined, the M0



M3	MOF (UML restriction)				
M2	UML metamodel	MASCARET metamodel			
M1	UML user model	VE1 model			...
M0	*user object*	*VE1a*	*VE1b*	...	...

Figure 8: *The multi-layer architecture of* MASCARET *(w.r.t* MOF *framework) for semantic modelling of VEs.*

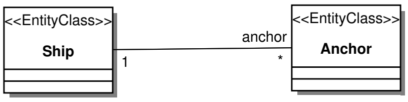
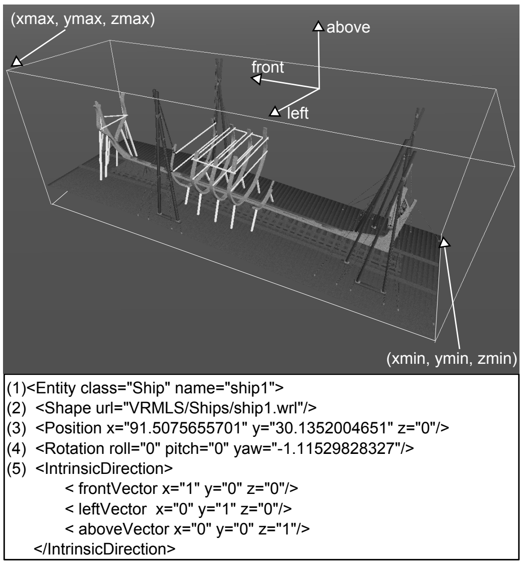


Figure 9: *A simplified class diagram in* MASCARET *representing ships and anchors and their relation.*



```
(1)<Entity class="Ship" name="ship1">
(2)  <Shape url="VRMLS/Ships/ship1.wrl"/>
(3)  <Position x="91.5075655701" y="30.1352004651" z="0"/>
(4)  <Rotation roll="0" pitch="0" yaw="-1.11529828327"/>
(5)  <IntrinsicDirection>
          < frontVector x="1" y="0" z="0"/>
          < leftVector  x="0" y="1" z="0"/>
          < aboveVector x="0" y="0" z="1"/>
     </IntrinsicDirection>
```

Figure 10: *Top: a ship in the application. Bottom: the definition and the instantiation of the ship in* MASCARET.

layer describes the instanciation of the VE. Several VEs can be instantiated from the same conceptual model. Fig. 10 illustrates the instanciation of a specific type of ship, a directional entity derived from the `Ship` class.

Finally, MASCARET provides meta-models of a VE at the M2 layer. A meta-model contains meta-data about concepts used in the M1 model (e.g., type, structure, states, operations an entity can realise; and its relations with others). Thus, the meta-model allows the reification of the conceptual model.

4.2. Modelling Directional Relations Using VRX-OCL

VRX-OCL stands for Virtual Reality eXtension of Object Constraint Language (OCL-http://www.omg.org/spec/OCL/). As a formal language, VRX-OCL intervenes in MASCARET at the M1 layer (i.e., the conceptual model) to specify dynamic constraints, notably spatial constraints in VEs. VRX-OCL allows a precise specification of constraints on any elements of the conceptual model of a VE represented by UML diagrams. Using the class model presented in Fig. 9, the constraint "*the height of every ship must not over 20 (meters)*" is expressed in VRX-OCL as follows:

```
context Ship inv:
self.height < 20
```

In this example, the constraint is applied in the `height` attribute of the `Ship` class. Other elements can be involved in a VRX-OCL expression such as operations, states, and associations. A constraint can be used to specify pre- or post-condition of an operation, or a guard condition for a state transition. Moreover, based on OCL, VRX-OCL allows the description of complex constraints such as cardinality constraints or logical constraints. The following example specifies that "*a ship must have at least one anchor*".

```
context Ship inv:
self.anchor->size() >= 1
```

This example illustrates a constraint over multi-classes. The navigation between classes (`Ship` and `Anchor`) is realized by means of an association between them (named as `anchor`). With regard to directional constraints, VRX-OCL is enriched by directional operators. Every operator corresponds to a relation defined previously in Sect.3. Considering a situation in Fig. 11, a directional constraint from a first-person perspective such as "*A worker, called worker A, is behind and left of the ship*" is expressed as follows.

```
context Ship inv:
let ship1:Ship = self.allInstances()
          ->select(name = 'ship1'),
    workerA:Worker = Worker.allInstances()
          ->select(name = 'workerA') in
 workerA.behindLeftOf(ship1)
```

Similarly, a directional constraint under a third-person perspective like "*From the viewpoint of the worker A, the anchor is on the left of the ship*" is expressed as follows.

```
context Worker inv:
let ship1:Ship = Ship.allInstances()
          ->select(name = 'ship1'),
    anchor1:Anchor = Anchor.allInstances()
          ->select(name = 'anchor1'),
    workerA:Worker = self.allInstances()
          ->select(name = 'workerA') in
 anchor1.leftOf(ship1)@viewpoint(workerA)
```

In this example, we extended the syntax of the original OCL by the `@viewpoint` operator that allows the definition of a deictic FoR.

5. Application

This section depicts how we used our directional model and the VRX-OCL language in a cultural heritage application named BrestCoz. We use the semantic model of direction to specify human-like behaviours of artificial agents and to visualise spatial constraints in BrestCoz. For a more detailed description of this application readers can refer to [BDIL*11]. BrestCoz is a VE for visiting Brest harbour (France) in 18th century that uses VR techniques to reconstruct a historical site and thus allow one to visit and specific shipbuilding activities (Fig. 11).

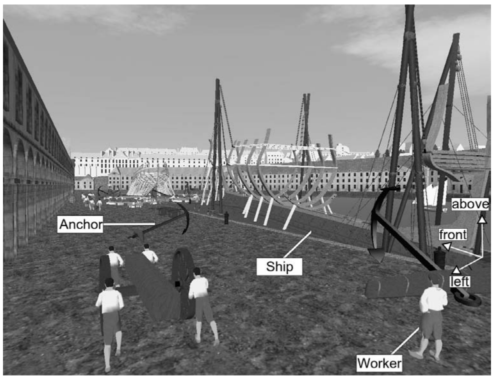


Figure 11: *A view in BrestCoz application (labels were added for illustrating the proposed directional model).*

5.1. Modelling Behaviours of Artificial Agents

BrestCoz is designed using Mascaret framework. Using a meta-modelling approach involved a conceptual modelling phase, Mascaret allows domain experts (marine historians), graphical designers, and software engineers to work together sharing their knowledge on a same model. In BrestCoz a virtual guide is able to explain to users a specific domain concept such as "*What is a keel in a ship?*" or a domain activity such as "*What does a carpenter do?*". Such semantic explanations are possible thanks to meta-models of Mascaret that allow a real-time reification and introspection of the domain model.

An important issue in BrestCoz is related to the modelling of domain activities such as shipbuilding. These domain-specific activities have been described at the conceptual level using Mascaret by means of Uml activity diagrams. An activity diagram depicts the procedure to realise a task, with reference to resources required and roles (in a hierarchical organisation). Domain activities are then simulated by artificial agents in interaction with users. Many behaviours of these agents are based on spatial constraints. For example, "*A bearer must carry timbers to put them in front of a carpenter*". Using Mascaret, it is possible to model the activity "carry", with the resources required as "timbers", and "a bearer" as the performer . Spatial expressions like "in front of a carpenter" are expressed using Vrx-Ocl. These constraints are used as a pre- or post-condition of an activity. The satisfaction of constraints allows to know whether an activity can be activated, and it is accomplished or not.

5.2. Visualising Spatial Constraints

As presented previously, in BrestCoz, users are free to visit the environment and discover domain activities. During their visit, it is quite often that users need helps to better localise an item in the space that is in a spatial relation with other items, for example to localise an anchor that is about 20m from the left of the ship. In BrestCoz, spatial constraints are expressed using Vrx-Ocl. Furthermore, when users can not localise the anchor, it is possible to visualise the area that contains the anchor based on predefined spatial constraints. That is, semantic areas such as "left", "right", or "behind" the ship can be visualised (see Fig. 12). Moreover, it is also possible to highlight the reference object (e.g., the ship), the primary object (i.e., the anchor), or the viewer.

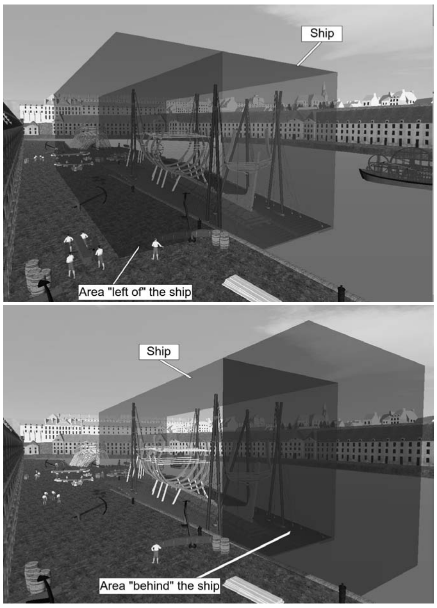


Figure 12: *Visualising spatial constraints in BrestCoz. Top: the area "on the left and 20 meters" from the ship. Bottom: the area "behind and 20 meters" from the ship.*

6. Conclusions and Perspectives

We introduced and integrated directional relationships as a new type of semantics into VEs. Our directional model enables the specification of semantic relations such as "left", "right" or "north", "south" in VEs. We proposed a formal model for direction in VEs. The model is based on a qualitative description of direction that has proven to be closer to how human represents the world. We defined a clear semantics of reference systems used for describing direction, from absolute (such as magnetic poles) to relative (such as intrinsic direction) reference systems. Our directional model allows to specify direction both from a first- and third-person perspective. Furthermore, we proposed a generic architecture and a spatial language named Vrx-Ocl for modelling direction in VEs. The architecture is based on a meta-modelling approach. The conceptual layer provides an abstract representation of a VE. The meta-model layer grants meta-access to information described in the conceptual level. Using VRX-OCL language, it is possible to specify constraints-based behaviours at the conceptual level.

This work can be extended in several directions. First, we plan to combine directional constraints with other types of spatial constraints such as topological, projective, and distance constraints presented in our previous work [TQDLC10]. [BG07] has shown that such a combination allows a more precise representation of space. Second, we plan to incorporate into our model suitable spatial reasoning techniques that enable artificial agents to find out new relationships from existing ones. Finally, another research could be related to further applications and cognitive validations of the integration of semantic spatial relationships into VEs. Taking a navigation aid system for example, such a future work might be on how the visualisation of spatial constraints can help users in navigating in large-scale VEs.

Acknowledgements. This work was supported in part by a grant from Région Bretagne (France) and in part by the ANR (Corvette project ANR-10-CORD-012).

References

[AC01] AYLETT R., CAVAZZA M.: Intelligent virtual environments - A state-of-the-art report. In *STAR Proc. of Eurographics 2001* (Manchester, UK, 2001), Eurographics Association. 1

[BC06] BILLEN R., CLEMENTINI E.: Projective relations in a 3D environment. In *GIScience* (2006), pp. 18–32. 2

[BDlL*11] BARANGE M., DE LOOR P., LOUIS V., QUERREC R., SOLER J., TRINH T.-H., MAISEL E., CHEVAILLIER P.: Get involved in an interactive virtual tour of brest harbour: Follow the guide and participate. In *Proceedings of the 11th International Conference on Intelligent Virtual Agents, IVA'11* (Reykjavick, Iceland, Septembre 2011). 6

[BG07] BRAGEUL D., GUESGEN H. W.: A model for qualitative spatial reasoning combining topology, orientation and distance. In *Proc. of the Twentieth Int'l Florida Artificial Intelligence Research Society Conf. (FLAIRS)* (May 7-9 2007), pp. 653–658. 8

[BNC*03] BOWMAN D. A., NORTH C., CHEN J., POLYS N. F., PYLA P. S., YILMAZ U.: Information-rich virtual environments: theory, tools, and research agenda. In *VRST'03: Proc. of the ACM symp. on Virtual Reality Soft. and Technol.* (2003), pp. 81–90. 1

[BR09] BORRMANN A., RANK E.: Specification and implementation of directional operators in a 3D spatial query language for building information models. *Advanced Engineering Informatics 23*, 1 (2009), 32–44. 3

[CB06] CLEMENTINI E., BILLEN R.: Modeling and computing ternary projective relations between regions. *IEEE Trans. on Knowl. and Data Eng. 18*, 6 (June 2006), 799–814. 2

[CLJZ07] CHEN J., LIU D., JIA H., ZHANG C.: Cardinal direction relations in 3D space. In *KSEM 2007: Proc. of the Second Int'l Conf. on Knowledge Science, Engineering and Management* (2007), vol. 4798 of *LNCS*, Springer, pp. 623–629. 3

[CR08] COHN A., RENZ J.: Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence 3* (2008), 551–596. 2

[CS10] CHEN T., SCHNEIDER M.: Modeling cardinal directions in the 3d space with the objects interaction cube matrix. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (New York, NY, USA, 2010), SAC '10, ACM, pp. 906–910. 3

[CTB*11] CHEVAILLIER P., TRINH T.-H., BARANGE M., DEVILLERS F., SOLER J., DE LOOR P., QUERREC R.: Semantic modelling of virtual environments using MASCARET. In *SEARIS@VR'11: Proc. of the 4th Workshop on Soft. Eng. and Architectures for Realtime Interactive Systems* (2011). 5

[CTCC99] CHEN C., THOMAS L., COLE J., CHENNAWASIN C.: Representing the semantics of virtual spaces. *IEEE MultiMedia 6*, 2 (1999), 54–63. 1

[DP02] DARKEN R., PETERSON B.: *Handbook of Virtual Environments*. Lawrence Erlbaum Associates, 2002, ch. 24, Spatial orientation, wayfinding, and representation, pp. 493 – 518. 2

[Fra92] FRANK A.: Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages and Computing 3* (1992), 343–343. 2, 3

[Fre92] FREKSA C.: Using orientation information for qualitative spatial reasoning. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Intl. Conf. GIS—From Space to Territory* (1992), vol. 639 of *LNCS*, Springer, pp. 162–178. 2

[Her94] HERNÁNDEZ D.: *Qualitative Representation of Spatial Knowledge*, vol. 804 of *LNAI*. Springer, Berlin, 1994. 3

[IDM06] IBÁÑEZ J., DELGADO-MATA C.: A basic semantic common level for virtual environments. *IJVR - Int'l J. of Virtual Reality 5*, 3 (2006), 25–32. 3

[KCM06] KALOGERAKIS E., CHRISTODOULAKIS S., MOUMOUTZIS N.: Coupling ontologies with graphics content for knowledge driven visualization. *IEEE VR* (2006), 43–50. 1

[KMDT*08] KLEINERMANN F., MANSOURI H., DE TROYER O., PELLENS B., IBÁÑEZ-MARTÍNEZ J.: Designing and Using Semantic Virtual Environment over the Web. *Int'l J. of Virtual Reality 7*, 3 (2008), 53–58. 1

[LBW05] LATOSCHIK M. E., BIERMANN P., WACHSMUTH I.: Knowledge in the loop: Semantics representation for multimodal simulative environments. In *Smart Graphics '05* (2005), pp. 25–39. 1

[PET02] PACHECO J., ESCRIG M. T., TOLEDO F.: Qualitative spatial reasoning on three-dimensional orientation point objects. In *16th Int'l WorkShop on Qualitative Reasoning (QR'02)* (Barcelona, Spain, 2002), pp. 113–124. 2

[SB04] SCHAFER W., BOWMAN D.: Evaluating the effects of frame of reference on spatial collaboration using desktop collaborative virtual environments. *Virtual Reality 7*, 3 (2004), 164–174. 3

[SLC99] SHEKHAR S., LIU X., CHAWLA S.: An object model of direction and its implications. *Geoinformatica 3*, 4 (1999), 357–379. 5

[STV09] SALAMIN P., THALMANN D., VEXO F.: Intelligent switch: An algorithm to provide the best third-person perspective in augmented reality. In *CASA 2009: Proc. of the 22nd Annual Conf. on Computer Animation and Social Agents* (2009). 3

[TQDLC10] TRINH T.-H., QUERREC R., DE LOOR P., CHEVAILLIER P.: Ensuring semantic spatial constraints in virtual environments using UML/OCL. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST)* (Hong Kong, 22-24 November 2010), pp. 219–226. 8

[WHK98] WALLER D., HUNT E., KNAPP D.: The transfer of spatial knowledge in virtual environment training. *Presence: Teleoper. Virtual Environ. 7*, 2 (1998), 129–143. 2

[WLGB00] WALLER D., LOOMIS J., GOLLEDGE R., BEALL A.: Place learning in humans: The role of distance and direction information. *Spatial Cognition and Computation 2*, 4 (2000), 333–354. 2