3D interaction assistance in virtual reality: a semantic reasoning engine for context-awareness

Yannick Dennemont, Guillaume Bouyer, Samir Otmane, Malik Mallem

3D interaction assistance in virtual reality: a semantic reasoning engine for context-awareness

Yannick Dennemont, Guillaume Bouyer, Samir Otmane, and Malik Mallem

IBISC laboratory, Evry university, France
{yannick.dennemont,guillaumme.bouyer,samir.otmane,malik.mallem}@ibisc.fr

Abstract. This work focuses on 3D interaction assistance by adding adaptivity depending on the tasks, the objectives, and the general interaction context. An engine to reach context-awareness has been implemented in Prolog+CG which uses Conceptual Graphs (CGs) based on an ontology. CGs descriptions of the available sensors and actuators in our scene manager (Virtools) allow the engine to take decisions and send them through Open Sound Control (OSC). Measurements and adaptations corresponding to specific tools uses are decided from rules handled by the engine. This project is a step towards Intelligent Virtual Environments, which proposes a hybrid solution by adding a separate semantic reasoning to classic environments. The first experiment automatically manages few modalities depending on the distance to objects, user movement, available tools, etc. Gestures are used both as an engine direct control and as an interpretation of user activity.

Key words: Interaction Techniques, Context-awareness, Knowledge Representation Formalism and Methods,Virtual reality

1 Interaction adaptation: toward context-awareness

Tasks in immersive virtual environments are associated to 3D interaction (3DI) techniques and devices (e.g. selection of 3D objects with a flystick using raycasting or virtual hand). As tasks and environments become more and more complex, these techniques can no longer be the same for every applications. A solution can be to adapt the interaction [5] to the needs and the context in order to improve usability, for example to:

- choose other techniques ("specificity") or make techniques variations ("flavor")[18];
- add or manage modalities[12][4][18];
- perform automatically parts of the task [7].

These adaptations can be done manually by the developer or the user, or automatically by the system: this is "adaptive" or "context-aware" 3DI . This open issue enables to:

- speed up the interaction [7];
- diminish the cognitive load (as in ubiquitous computing);
- tailor the interaction [22] [18];
- add or manage interaction possibilities [4].

In order to go beyond basic interaction, adaptive systems can first provide recognitions from raw data (on an activity recognition layer, Figure 1). But to achieve a better adaptivity, more content is needed: the context. A formal and well recognized definition is [10]: *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.* Thus, an ideal system for 3DI assistance is context-aware as *it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*

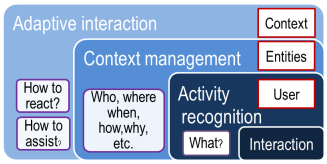


Fig. 1: Different layers to reach adaptive interaction

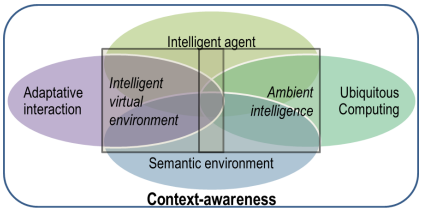


Fig. 2: Different families of context-aware applications

Context-awareness emerged from intelligent systems [6]. Some drawbacks were due to fully abstract reasoning or user exclusion. Intelligent assistance systems can be split in two trends. Systems tend to stress user assistance on well defined context (e.g. [4]) or to stress context identification that leads to direct adaptations for each situation (e.g. [9][11]). Context-awareness has different focuses (Figure 2), yet there is a shared ideal list of properties to handle [1]:

- Heterogeneity and mobility of context;
- Relationships and dependencies between context;
- Timeliness: access to past and future states;
- Imperfection: data can be uncertain or incorrect;
- Reasoning: to decide or to derive information;
- Usability of modelling formalisms;
- Efficient context provisioning.

Examples	Representation	Reasoning	Semantic Approach	Uncertainty degree	Representation modification	Reasoning modification	Usability	Aims
SOCAM [13]	OWL (Web Ontology language)	FOL+Bayesian	Yes	Yes	High	High	Medium	Middleware for ubiquitous computing
GAIA [24]	FOL (First Order Logic)	FOL	Yes	Yes	High	High	Medium	Ubiquitous computing services
VR-UCAM [18]	5W1H tuples (Who, What, When, Where, Why, How)	Condition Matching	No	No	Low	Medium	High	Extend ubiquitous computing services to VR
VR-DeMo [20]	MBUID (model based user interface design)	Event Condition Action	No	No	Medium	Medium	High	Automatic personalized 3D interaction
Our Engine	CG (Conceptual Graphs)	CG theory, FOL	Yes	Yes	High	High	High	Context-aware services for VR. Applications to adaptive 3D interaction.

Fig. 3: Approaches comparisons

Our research is mainly in the adaptive 3D interaction field. Yet, to achieve wider and better 3DI, a richer context with semantic information and/or in-

telligent agents is needed. Also reasoning needs will grow with the available information. So our approach is generally part of the Intelligent Virtual Environments. Adaptive 3DI can be implicit with adaptations embedded in the interaction techniques [20][3], or explicit by using external processes [16][7][4][18]. Some frameworks are generic enough (examples and their comparison on Fig. 3) but not able to describe any situations, to modify their reasoning or difficult to reuse/to expand (particularly when thought for another domain).

– which performs semantic reasoning through logical rules on an ontology;
– which communicates with application tools: sensors to retrieve the context, and actuators to manage visual, audio and haptic modalities as well as interaction modifications;
– which is generic: pluggable to existing non-semantic virtual environment if tools are available.

Users will benefit from an automatic 3D interaction assistance that can supply support through modalities, interaction technique choice or application-specific help depending on the current situation. Besides, designers could reuse, rearrange and modify this 3DI adaptivity to share reasoning between applications or to create application-specificity. A good adaptive 3DI can also help to release the designers from the prediction of every situations, thus it should be able to deal with degree of unpredictability. Next, we discuss our choices for modelling context and reasoning to achieve these goals.

2 Knowledge Representation and Reasoning

We need to manage context and to decide how to react, which is a form of Knowledge Representation and Reasoning. More precisely, our system needs first to retrieve and represent items of information, possibly specific to an application, then to handle this context and to define its effects on 3DI (discussed by [11] for virtual reality). Several criteria led our choice for the engine core: semantic degrees, expressiveness (vs efficiency) and usability. We choose to base our representation on Conceptuals Graphs (CGs). They have a strong semantic founding and are built on an ontology. They provide a good expressiveness (a universal knowledge representation [21][8]) equivalent to First Order Logic (FOL) but with a better usability since they are also human readable. FOL is usually the most expressive choice made for context-awareness. Meantime, semantic reasoning with an ontology is the most used approach in context-awareness as it provides interoperability and a non-abstract representation. Moreover coupled with the CGs usability, the model may allow at some point a welcomed direct users involvement [6]. Semantic virtual worlds as a new paradigm is a discussed issue [15]. Several approaches offer to build full semantic worlds [14], often with semantic networks [19][17][2] which reinforce our conviction for CGs. However we will not try to build a full semantic world but to gather semantic information to help the 3DI. We aim at context-awareness in classic applications with an external representation and reasoning engine.

3 Overview of the engine

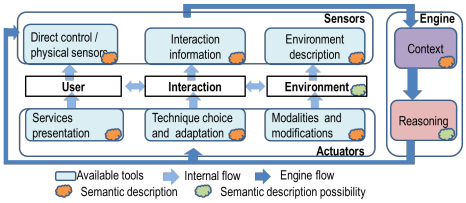


Fig. 4: An external engine - communication through semantic tools

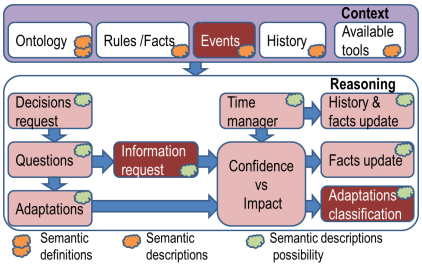


Fig. 5: The engine - forms of context and reasoning

The engine uses rules to take decisions regarding a stored context (knowledge, events etc.). Context and decisions concern the user, the interaction and the environment, which communicates with the engine through different tools (Fig. 4). Tools must have a semantic description of their uses in order to be triggered by the engine. They can be actuators with perceivable effects or sensors that retrieve information. Those tools can embed other forms of reasoning than the engine core (e.g Hidden Markov Models) to provide information.

Context have various forms managed by the decision process(Fig. 5). First, the ontology lists concepts and relations with underlying semantic, which are used by CGs in order to describe rules and facts. Available tools and the past events in history are special facts. Events are newly integrated information and trigger a decision request in an automatic mode. The time manager checks the validity of the needed facts. When a decision with an associated tool is true, the engine aggregates its confidence and impact from facts, events' timing and rules. An acceptable total impact induces a knapsack problem as a last classification.

We use Virtools as our scene graph manager and the Amine platform [13] (a Java open-source multi-layer platform for intelligent systems) for the engine. This platform offers an ontology manager and a FOL backward chaining engine that handles CGs: Prolog+CG (PCG). Open Sound Control protocol (OSC) is used for communication between the scene and the engine.

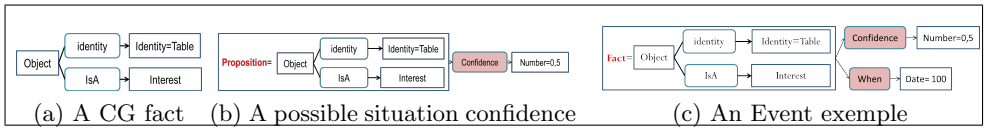
4 Concepts and conceptual graphs in the engine

The focus of the engine is to be easily modifiable, reusable, and expanded by designers and users. Therewith, we want to reason with ideas and situations rather than formulas. This is where the ontology is important as it defines our semantic vocabulary (written in italic afterwards). Situations can now be described using CGs built with these concepts and classified using CGs theory. In a final form, every logical combinations in a CG (that a user could enter) should be handled. And as the engine is used, the ontology is refined. This in order to allow simpler rules, easier to reuse reasoning as long as a better concepts classification. Next is a rough taxonomy of currently used concepts. We need to be able to express:

- **Reasoning concepts**: state what is true (*fact*) and what is just a matter of discussion (*proposition*); *rules* (*effects* depending on *causes*); degree of *confidence* in those concepts (e.g. Fig. 6 and Fig. 7). Also what decisions can actually be made (*reactions* like *adaptations* or *questions*, e.g. Fig. 8), etc.;
- **Reification concepts**: Manage *tools*, like *sensors* or *actuators*. Descriptions include *commands* to be sent for specific *uses* and their *impacts* (e.g. Fig. 9) depending on *cases* (e.g. Fig. 10);
- **3D interaction concepts** : the main focus of the overall generic engine. So we need to describes various *modalities*, *tasks* etc. For example, part of the user *cognitive load* is linked to the total amount of *impact* used;
- **Time concepts**: manage new *facts*, events (*fact* with a *date* and a *duration*) (e.g. Fig. 6), etc. *History* manage previous event, *reactions*, etc. ;
- **Spatial concepts**: manage *position*, *direction* etc. In virtual environment, a lot of the spatial issues are in fact handled by the scene graphs manager. But *zones* like *auras* or *focuses* are useful to understand the current activity.
- **General concepts**: base vocabulary to describe situations. For example to manipulate *attributes* like *identity* or express *active* states.
- **Application specific concepts** : applications can expand the knowledge base with their own concepts. For example *gestures* that can be named ('Z'), and/or classed (*right* and *up* are also *rectilinear* gestures).

5 Reactions process

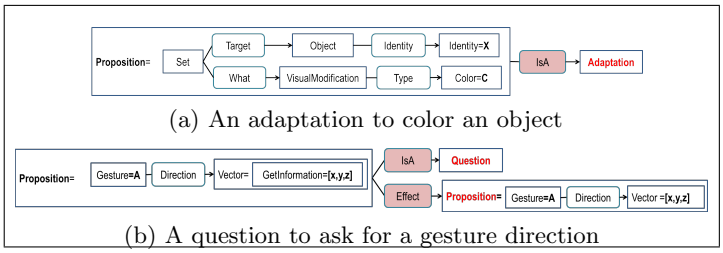
PCG is a backward chaining engine. Thus it can answer if specific facts can be inferred or not. A meta-interpreter has to be written to do forward-chaining, i.e to list what can be deducted given the available facts. Our meta-interpreter do both to manage forms of truth (as a PCG element, as a *fact* description,



(a) A CG fact (b) A possible situation confidence (c) An Event exemple

Fig. 6: Facts examples: about interest, confidence and event

Fig. 7: Rule example: the enhancement will of an interest



(a) An adaptation to color an object

(b) A question to ask for a gesture direction

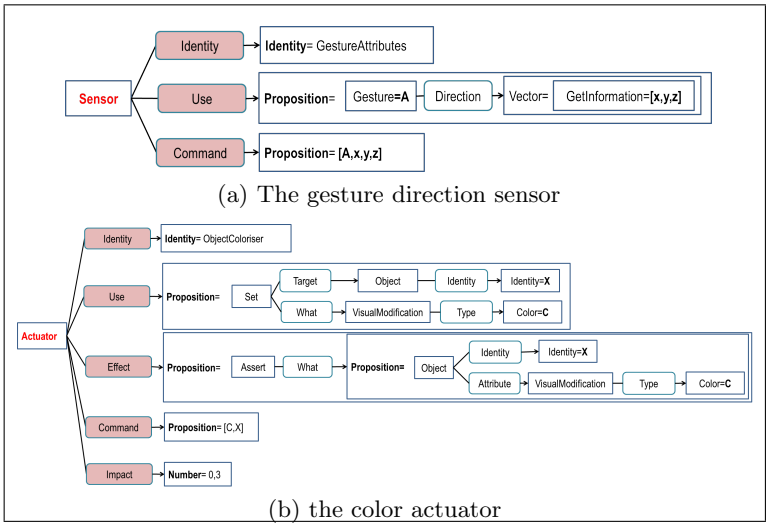
Fig. 8: Reaction possibilities examples

as a CG *rules effect* etc.), degree of truth (*confidence*) and times (*duration validity*, *history* etc.). At any time, the engine store context elements (*facts*, *events*, etc.). When an application need a fitting *reaction* (after a new *event*, when ordered by the user, etc.), it send a decision request. The engine then use the meta-interpreter to seek eligible *reactions*. Those are *true adaptations* and *questions* (e.g Fig. 8) with an available associated *tool* (e.g. Fig 9). Then the engine aggregates decisions' *confidence*. A list of *confidence* is obtained by considering all paths leading to the *reactions*. Each path can combined different *confidence* expressions:

- A direct PCG fact (e.g. Fig 6a) has the maximum confidence: 1;
- A *fact* confidence or a generic knowledge confidence (e.g. Fig. 6b);
- Event confidence (e.g. Fig. 6c). A *fact* confidence, but time dependant as the initial supplied confidence is multiplied by the ratio of remaining validity.
- CGs rules induced confidence (e.g. Fig. 7). If true, the effects confidences are the average causes confidences times the rule confidence (0 instead). It is an iterative process.

We use a fusion function to convert this list into a single scalar. We consider that the more facts and rules led to a reaction, the more the confidence in it should increase, while kept bounded between 0 and 1. So for n confidences with *Mean* as average value: $Globalconfidence = (1 - Mean) \times (1 - \frac{1}{n}) \times Mean + Mean$. The global confidence remain 0 (respectively 1) in case of absolute false when $Mean = 0$ (respectively absolute true, $Mean = 1$). Singleton is not modified.

Next, the engine aggregates the decisions impact. Each tool has an initial impact which is modified by specific cases. E.g Fig. 10, the impact of a de-



(a) The gesture direction sensor

(b) the color actuator

Fig. 9: Tools examples

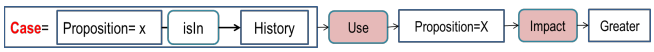


Fig. 10: impact increase case example: to avoid activation/reactivation cycle

cision already in the history increases (to reduce activation/deactivation cycles). Initial impact equals to 0 (without any impacts) or 1 (the most impacting) are unmodified. Otherwise, at each n applicable case, the impact is altered with a weight (W, 25% if not supplied) while kept bounded: $impact(n) = impact(n-1) + W \times (1 - impact(n-1))$ for *greater* impacts or $impact(n) = impact(n-1) - W \times impact(n-1)$ for *lower* impacts. Thus smaller steps are made for already extreme values (e.g. keeping impacting situations reachable).

Finally, decisions with a confidence on impact ratio greater than a threshold (1 by default) are eligible. Then, eligible decisions are selected to fill the available admissible impact. Thus this last classification is a knapsack problem. The available impact is the initial user impact (a first step into profiling the user) minus the active decisions impact.

6 Scenario and case study examples

We test the engine with a case study: to try to automatically acquire some user's interests and enhance them. The interests are here linked to the user's hand. The application tools are:

- a Zones Of Interest (ZOI) sensor to add and report the content of 3D zones;
- a gestures recognition and a gestures attributes sensors (Fig. 9a).
- a sensor of hand movement speed and scope. Movement speed is qualified as high or low and movement scope as local or global;
- an actuator to change the color of an object (Fig. 9b);
- an actuator to add a haptic or a visual force to an object;

The engine uses general rules like:

- Define what is an interest (e.g is in a ZOI or is a previous interest);
- Try to enhance an interest (Fig. 7);
- Define possible enhancements: e.g object visual modifications through color change or interaction modifications through force (visual or haptic);
- General and adaptations states management:
  - remove visual modification for past interests;
  - remove a currently applied force if the movement is abnormal (e.g local+high=the user is "stuck").
- increase decision impact for some concepts (e.g haptic impact> visual impact and interaction modification impact > visual modification impact);
- increase decision's impact if present in history (Fig. 10);
- decrease interaction modification's impact for local movement.

Finally, the application specific rules are:

- Monitor the hand movement;
- Ask for the gesture attributes if a gesture is detected;
- Activate or deactivate a ZOI around the hand if a circular gesture occurred;
- Activate a ZOI in the direction of the gesture if a rectilinear gesture occurred;
- Deactivate this direction ZOI after 3s.

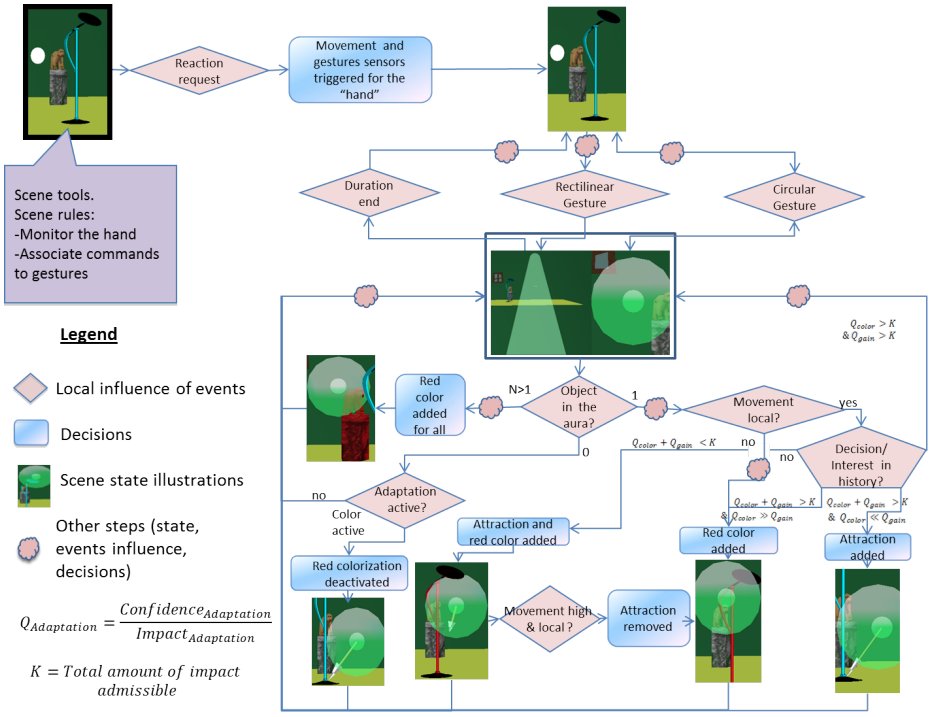


Fig. 11: The engine automatically apply adaptations depending on the context

– Deactivate every adaptations if the "Z" gesture is detected.

As a result, the rules combine themselves as expected (adaptations examples Fig. 11), but with supplementary outcomes which were not fully planned. In fact, there are several interests: explicit, by creating voluntarily a ZOI, or implicit, either by moving rectilinearly toward an object (thus creating a ZOI) or with previous interest. With a ZOI around the hand activated, passing by an object colors it red, while standing next to it makes it also attractive (as movement is then local, diminishing the attraction impact). Colors are reset when the user moves far away. Depending on the color actuator impact and the history, the coloration time can vary, and can even flash for a while (not intended at first). When pointing an object or when moving toward an object during a global movement, the object is colored red. When pointing an object from a rest position or starting a new movement directly toward an object, it makes it also attractive (not intended at first, this primary intention can be highlighted due to the latency of the movement scope sensors, which still point the movement as local). Colors are reset after this ZOI deactivation. In both cases, when pointed several times as an interest (thus present as several current facts or history facts) attraction can be activated regardless of the movement scope (global or local). Attraction is removed when the user tries to resist it. When it has been deactivated, gain usually cannot be reactivated for a time corresponding to history memory. Some reactivations occur for coloring as the decision has initially less impact. More

complex situations occur when several objects are close to the hand: e.g only the less impacting adaptation (coloring) is applied to a maximum of objects (even if for now there is no specific treatment for groups, the most fitting adaptations is applied until there is no more cognitive load thus a group logic emerged). Those results depend on the initial impact, confidence and cognitive load values.

7 Conclusion

The engine aims to allow a semantic reasoning and the reuse of tools in a non-semantic environment to help the 3DI. We propose an engine core with a semantic base to achieve adaptation, which could be directly addressed by designers or users. Context-awareness properties (page 2) are almost all tackled but need deepening. The engine response delay is not well suited for a full automatic mode yet, but rather for punctual helps. This drawback can be lessen but is an inherent part of our approach. By adding a planning block later, we could refine the adaptations and allow more tools combinations. Indeed active decisions could be replaced by better ones in a new context. This currently can be done by freeing cognitive load and rethink all adaptations at each decision process. However, it is a particular case of a more complex resources and world states planning. Besides, we have started adding direct control of the engine. This part emphasizes the engine tailorability since changing the control from a gesture to another, or to any events, can easily be done directly into the engine (rather that remodelling application parts). And as a 3DI rules set can be reused, any applications can add their own rules set and controls possibilities. Also, the gestures recognition can be used to monitor the user activity and to deduce hints of intention. A possibility is to use our HMM recognition module on different data to learn and classify interesting situations. Our next step is to continue to explore context (especially user intention hints) and adaptations for virtual reality with their implementation using tools and the engine. This work is supported by the AP7 DigitalOcean project.

References

1. C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, Apr. 2010.
2. B. Bonis, J. Stamos, S. Vosinakis, I. Andreou, and T. Panayiotopoulos. A platform for virtual museums with personalized content. *Multimedia Tools and Applications*, 42(2):139–159, Oct. 2008.
3. P. Boudoin, S. Otmane, and M. Mallem. Fly Over , a 3D Interaction Technique for Navigation in Virtual Environments Independent from Tracking Devices. *Virtual Reality International Conference*, (Vric), 2008.
4. G. Bouyer, P. Bourdot, and M. Ammi. Supervision of Task-Oriented Multimodal Rendering for VR Applications. *Eurographics Symposium on Virtual Environments*, 2007.
5. D. A. Bowman, J. Chen, C. A. Wingrave, J. F. Lucas, A. Ray, N. F. Polys, Q. Li, Y. Haciahmetoglu, J. sun Kim, S. Kim, R. Boehringer, and T. Ni. New Directions in 3D User Interfaces. *International Journal of Virtual Reality*, 5:3–14, 2006.

6. P. Brzillon. From expert systems to context-based intelligent assistant systems: a testimony. *The Knowledge Engineering Review*, 26(1):19–24, 2011.
7. A. Celentano and M. Nodari. Adaptive interaction in Web3D virtual worlds. *Proceedings of the ninth international conference on 3D Web technology - Web3D '04*, 1(212):41, 2004.
8. M. Chein and M. Mugnier. *Graph-bases Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer, 2009.
9. P. Coppola, V. D. Mea, L. D. Gaspero, R. Lomuscio, D. Mischis, S. Mizzaro, E. Nazzi, I. Scagnetto, and L. Vassena. *AI Techniques in a Context-Aware Ubiquitous Environment*, pages 150–180. 2009.
10. A. Dey and G. Abowd. Towards a better understanding of context and context-awareness. In *CHI 2000 workshop on the what, who, where, when, and how of context-awareness*, volume 4, 2000.
11. S. Frees. Context-driven interaction in immersive virtual environments. *Virtual Reality*, 14(4):277–290, Dec. 2010.
12. S. Irawati, D. Calderón, and H. Ko. Semantic 3D object manipulation using object ontology in multimodal interaction framework. In *Proceedings of the 2005 international conference on Augmented tele-existence*, pages 35–39. ACM, 2005.
13. A. Kabbaj. Development of intelligent systems and multi-agents systems with amine platform. In *Conceptual Structures: Inspiration and Application*, volume 4068 of *Lecture Notes in Computer Science*, pages 286–299. Springer Berlin / Heidelberg, 2006.
14. M. E. Latoschik, P. Biermann, and I. Wachsmuth. Knowledge in the Loop : Semantics Representation for Multimodal Simulative Environments (2005). pages 25 – 39, 2005.
15. M. E. Latoschik, R. Blach, and F. Iao. Semantic Modelling for Virtual Worlds A Novel Paradigm for Realtime Interactive Systems ? In *ACM symposium on Virtual reality software and technology*, 2008.
16. S. Lee, Y. Lee, S. Jang, and W. Woo. vr-UCAM: Unified context-aware application module for virtual reality. *Conference on Artificial Reality*, 2004.
17. J.-l. Lugrin and M. Cavazza. Making Sense of Virtual Environments : Action Representation , Grounding and Common Sense. In *12th International conference on intelligent user interfaces*, 2007.
18. J. Octavia, K. Coninx, and C. Raymaekers. Enhancing User Interaction in Virtual Environments through Adaptive Personalized 3D Interaction Techniques. In *UMAP*, 2010.
19. S. Peters and H. E. Shrobe. Using Semantic Networks for Knowledge Representation in an Intelligent Environment. In *1st International Conference on Pervasive Computing and Communications*, 2003.
20. I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80. ACM, 1996.
21. J. F. Sowa. Chapter 5 conceptual graphs. In V. L. Frank van Harmelen and B. Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 213 – 237. Elsevier, 2008.
22. C. A. Wingrave, D. A. Bowman, and N. Ramakrishnan. Towards preferences in virtual environment interfaces. In *Proceedings of the workshop on Virtual environments 2002*, EGVE '02, pages 63–72, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.