3D Interaction Assistance in Virtual Reality: A Semantic Reasoning Engine for Context-awareness

From Interests and Objectives Detection to Adaptations

Yannick Dennemont, Guillaume Bouyer, Samir Otmane and Malik Mallem
IBISC Laboratory, Evry University, 40 rue du Pelvoux, 91020 Courcouronnes, France

Keywords: Interaction Techniques, Virtual Reality, Context-awareness, Intelligent Systems, Knowledge Representations.

Abstract: This work aims to provide 3D interaction assistance in virtual environments depending on context. We designed and implemented a generic decision engine that can connect to our existing virtual reality applications through a set of tools. It uses an ontology and Conceptual Graphs (CGs) to represent knowledge, and First Order Logic to conduct semantic reasoning. Context information are gathered by virtual sensors in the application and interpreted by the engine. Multimodal assistance is provided by virtual actuators. Our first test scenario is about assistance to selection of objects or navigation towards objects: the engine automatically detects user's interests and manages adaptations depending on user's hand gestures, interactions history and type of task.

1 INTRODUCTION

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 (Bowman et al., 2006) to the needs and the context in order to improve usability, for example:

- to choose other techniques (”specificity”) or make techniques variations (”flavor”)(Octavia et al., 2010);
- to add or manage modalities(Irawati et al., 2005)(Bouyer et al., 2007)(Octavia et al., 2010);
- to perform automatically parts of the task (Celentano and Nodari, 2004).

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 (Celentano and Nodari, 2004);
- diminish the cognitive load (as in ubiquitous computing);
- tailor the interaction (Wingrave et al., 2002) (Octavia et al., 2010);
- add or manage interaction possibilities (Bouyer et al., 2007).

In order to go beyond basic interaction (Fig. 1), adaptive systems can first provide recognitions from raw data. Usually they focus on the user and an activity recognition layer. But to achieve a better adaptivity, more content is needed: the context. The context regroups information from potentially every entities and can be used by the application to react and assist the interaction. A formal and well recognized definition is (Dey and Abowd, 2000): *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.*

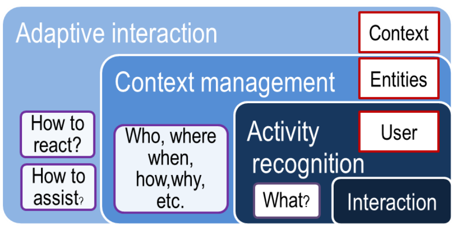


Figure 1: Different layers to reach adaptive interaction.

Dennemont Y., Bouyer G., Otmane S. and Mallem M..
3D Interaction Assistance in Virtual Reality: A Semantic Reasoning Engine for Context-awareness - From Interests and Objectives Detection to Adaptations.
DOI: 10.5220/0004207503490358
In *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications* (GRAPP-2013), pages 349-358
ISBN: 978-989-8565-46-4

2 RELATED WORK

Context-awareness emerged from intelligent systems (Brézillon, 2011). 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. (Bouyer et al., 2007)) or to stress context identification that leads to direct adaptations for each situation (e.g. (Coppola et al., 2009)(Frees, 2010)). Context-awareness has different focuses (Fig. 2), though there is a shared ideal list of properties to handle (Bettini et al., 2010):

- 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.

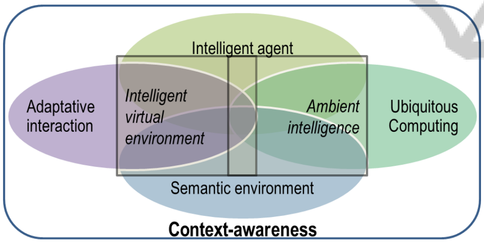


Figure 2: Different families of context-aware applications.

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 intelligent agents is needed. Also reasoning needs 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 (Poupyrev et al., 1996)(Boudoin et al., 2008), or explicit by using external processes (Lee et al., 2004)(Celentano and Nodari, 2004)(Bouyer et al., 2007)(Octavia et al., 2010). Semantic virtual worlds can be considered as a mixed form: explicit processes can be applied but they are directly embedded into the rendering loop for the specific semantic description of the environment. Semantic virtual worlds as a new paradigm is a discussed issue (Latoschik et al., 2008). Several approaches offer to build full semantic worlds (Latoschik et al., 2005)(Peters and Shrobe, 2003)(Lugrin and Cavazza, 2007)(Bonis et al., 2008). Ubiquitous computing offers a lot of frameworks for reaching context-awareness (Dey et al., 2001)(Ranganathan and Campbell, 2003)(Gu et al., 2004)(Coppola et al., 2009).

Finally how can we achieve a generic interaction assistance for virtual reality? Firstly, we want to be able to describe a generic assistance. Therefore we can not base our project on works that focus on implicit adaptations, on very specific assistances (command disambiguation (Irawati et al., 2005), recognizing and doing part of the task for the user (Celentano and Nodari, 2004)), or on one emphasized aspect of assistance (personalisation (Octavia et al., 2010)). Secondly, we will not try to build a full semantic world but to gather semantic information to help the 3DI. This will allow the assistance to be used by classic applications (which are the most common for now). Thirdly, we want to be able to both identify a general context and to modulate our reasoning (thus our assistance). That limits the reuse of previous work stressing strongly only one aspect ((Bouyer et al., 2007)(Coppola et al., 2009)(Frees, 2010)). Finally, 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 they were thought for another domain). The choice of context representation can be restrictive: low level key-values (Dey et al., 2001); a fixed list of data pairs (Lee et al., 2004) or a first order logic predicate with a tuple argument (Ranganathan and Campbell, 2003) (which

Examples	Representation	Reasoning	Semantic Approach	Uncertainty degree	Representation modification	Reasoning modification	Usability	Aims
SOCAM (Gu et al., 2004).	OWL (Web Ontology language)	FOL+Bayesian	Yes	Yes	High	High	Medium	Middleware for ubiquitous computing
GAIA (Ranganathan and Campbell, 2003)	FOL (First Order Logic)	FOL	Yes	Yes	High	High	Medium	Ubiquitous computing services
VR-UCAM (Lee et al. 2004).	5W1H tuples (Who, What, When, Where, Why, How)	Condition Matching	No	No	Low	Medium	High	Extend ubiquitous computing services to VR
VR-DeMo (Octavia et al., 2010)	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.

Figure 3: Approaches comparisons.

limits the ease to express complex relations and situations). Ontological approaches have richer and wider range of representation but do not offer the same possibilities for reasoning modification(Gu et al., 2004). Thus, part of the reasoning is usually done by a second method (often first order logic).

To sum up, this research aims to model and develop an explicit semantic context-aware engine for common 3DI which:

- is generic; can represent any context and reactions;
- is usable, extensive and modifiable; performs semantic reasoning with logical rules on an ontology;
- is pluggable; communicates with application tools: sensors to retrieve the context, and actuators to manage visual, audio and haptic modalities as well as interaction modifications.

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.

In the next section, we discuss our choices for modelling context and reasoning to achieve these goals. Afterwards the section 4 gives an overview of the whole engine. The section 5 and 6 respectively details the representations possibilities and the reasoning process to obtain automatic adaptations. The section 7 presents a test scenario with examples of adaptations and process parts. Finally the section 8 details our conclusion and perspectives.

3 REPRESENTATION AND REASONING

We need to manage context and to decide how to react, which is a form of Knowledge Representation and Reasoning. Actually, our system needs first to retrieve and represent items of information, then to handle this context and to define its effects on 3DI (discussed by (Frees, 2010) 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 (Sowa, 2008)(Chein and Mugnier, 2009)) equivalent to First Order Logic (FOL) but with a better usability since they are also human readable. Semantic networks are often picked to build the full semantic world (Peters and Shrobe, 2003)(Lugrin and Cavazza, 2007)(Bonis et al., 2008) which reinforces our conviction for CGs.

The needed expressiveness is an open issue yet *You Can Only Learn What You Can Represent* (Otterlo, 2009). Thus, it is a fundamental question for a sustainable use. 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 (Brézillon, 2011). Therefore using CGs, we obtain re-usability and interoperability (ontological approach), sustainability and generality (FOL expressiveness) and the usability (graphs representation).

4 OVERVIEW OF THE ENGINE

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 a set of 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 (environment or interaction modifications, services presentation etc.) or sensors that retrieve information (from the environment, by monitoring the interaction or through direct information from the user etc.). Those tools can embed other forms of reasoning than the engine core (e.g Hidden Markov Models) to provide information. Finally, tools can also apply to the engine itself. Meta-actuators, which have a perceivable effect on the engine, are currently used (parameters or rules modifications etc.). Meta-sensors could be used to call external reasoning possibilities (e.g. data treatment).

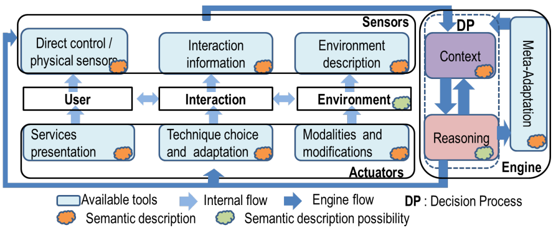


Figure 4: An external engine - communication through semantic tools.

Thus the engine offers meta-adaptations possibilities (it can modify itself depending on the context).

Context has various forms managed by the decision process (Fig. 13). 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 limits the decisions that can be made, which induces a knapsack problem as a last classification.

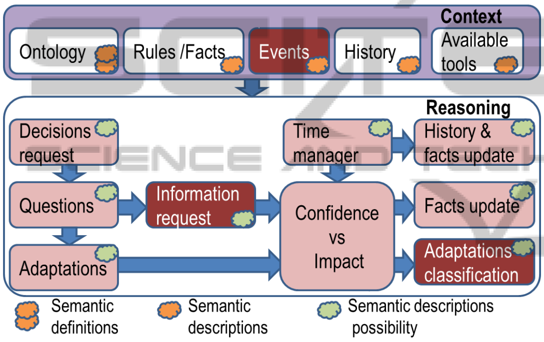


Figure 5: The engine - forms of context and reasoning.

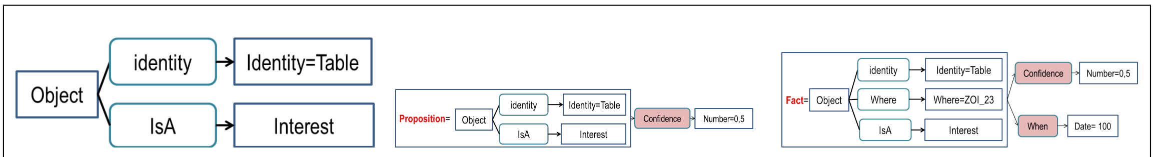
We use Virtools as our scene graph manager and the Amine platform (Kabbaj, 2006) (a Java open-source multi-layer platform for intelligent systems) for the engine. It offers an ontology manager and FOL backward chaining that handles CGs: Prolog+CG (PCG). Open Sound Control protocol (OSC) is used for communication between the scene and the engine.

5 CONCEPTS USE IN THE ENGINE

5.1 The Ontology

The specification 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). Next is a rough taxonomy of currently used concepts.

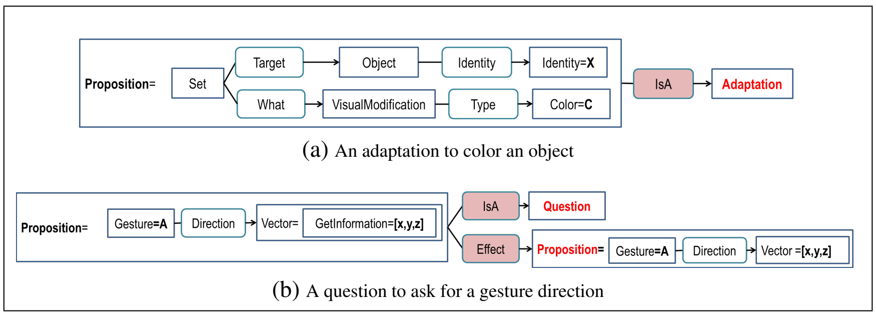
Reasoning Concepts. They are used to state what is true (*fact*) and what is just a matter of discussion (*proposition*); *rules* (*effects* depending on *causes*); degree of *confidence* in those concepts (e.g. Fig. 6 and Fig. 7). But also what decisions can actually be made (*reactions* like *adaptations* or *questions*, e.g. Fig. 8);



(a) CG fact (b) A general confidence (c) Event with confidence and time

Figure 6: Facts examples.

Figure 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

Figure 8: Reaction possibilities examples.

Reification Concepts. They are used to 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. They are the main focus of the overall generic engine. So we need to describe various *modalities*, *tasks* etc.

Time Concepts. They are used to manage new *facts*, events (*fact* with a *date* and a *duration*) (e.g. Fig. 6), *history*of previous events and *reactions*.

Spatial Concepts. They are used to 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. They are a 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).

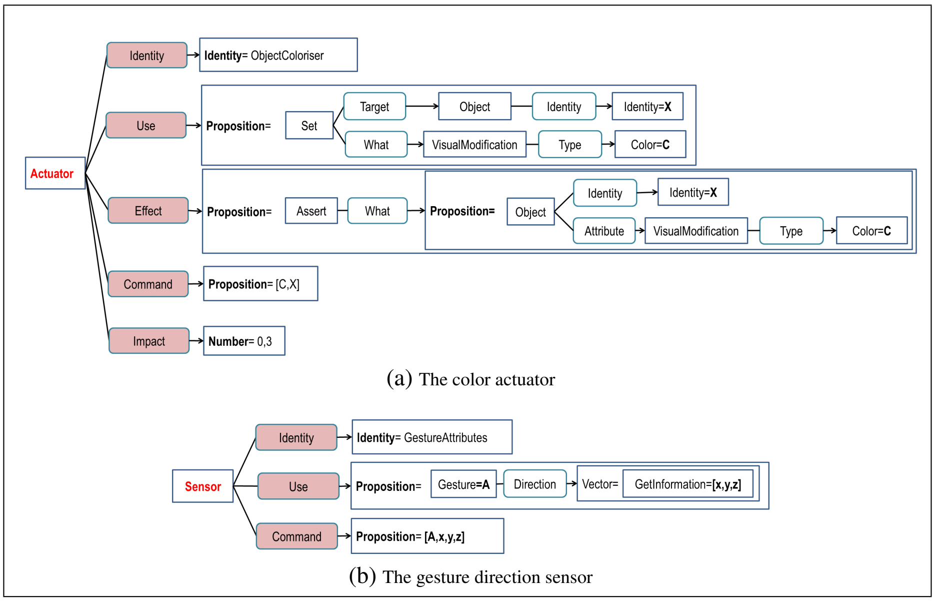
Let focus more on two concepts used to classify reactions (detailed in the section 6). We use *confidence* that represents the degree of sureness of an information. For a *reaction*, it thus reflects the degree of certainty that this decision can be applied in the current situation. *Impact* is specific to *reactions* and represents the degree of their perceived repercussion for the user. The initial value of impact is supplied by the tool used to reify the reaction (an intrinsic degree of repercussion). Then the actual impact can be modified depending on the current situation, which leads to the expression of influential *cases*.

5.2 Conceptual Graphs

Situations and reactions to situations can now both be described using CGs, built on those concepts and classified using CGs theory. In a final form, every logical combinations in a CG (that a user could enter) should be handled. Next are some examples illustrating different categories of context and reasoning:

Facts. Fig. 6 represents different information regarding a similar situation, about an *object* of *interest* with the *identity* "table". On figure 6a the situation is currently true. Figure 6b is an expression of the general *confidence* about this type of situations (without assuming its realization). Fig. 6c is an event (send by a *sensor*) with a *date* and an initial *confidence*.

Rules. Fig. 7 represents a general *rule*: if an *object* is a known *interest* then the engine will try to *enhance*



(a) The color actuator

(b) The gesture direction sensor

Figure 9: Tools examples.

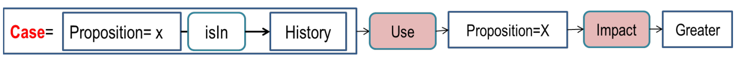


Figure 10: Impact increase case example: to avoid activation/reactivation cycle.

this *object*. This *rule* is associated with a high *confidence*; the situation is most probably true regardless of the remaining context.

Reactions. Fig. 8a is an example of general *adaptation* describing the action of modifying the *color* of an *object*, which is a type of *visual modification*. Fig. 8b is an application specific *question* to obtain the *direction* of a *gesture*.

Tools. Fig. 9a and Fig. 9b are *tools* able to implement the previous *reactions*. In fact, their descriptions of *use* are here exactly the *reaction* descriptions (they are relatively general tools). Nevertheless they could be more specific (e.g. a special tool to color a specific subtype of objects etc.). *Impact* of the color *actuator* is low (compared to an attraction for example) and the *impact* of the *sensors* is null (as completely transparent for the user).

Cases E.g Fig. 10 shows a situation that can modify the *impact* of *reactions*: the *impact* of a decision already in the *history* will increase. This will help to avoid activation/deactivation cycles.

6 REACTIONS PROCESS

How are those concepts and conceptual graphs used by the engine to obtain fitting reactions? They are handled by our Prolog meta-interpreter. It uses concepts definitions to be able to deal with forms of truth (as a PCG element, as a *fact* description, as a CG *rules effect* etc.), degree of truth (*confidence*) and times (*duration validity*, *history* etc.). At any time, the engine stores context elements (*facts*, *events*, etc.). When an application needs a fitting *reaction* (after a new *event*, when ordered by the user, etc.), it sends a decision request. The engine then uses 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).

In order to classify those decisions, we calculate both the *confidence* and the *impact* of each decision. For any CGs, and thus also for *reactions*, a list of *confidence* is obtained by considering all paths leading to them. Each path can combine different *confidence* expressions:

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

We use a confidence 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 is still 0 (respectively 1) in case of absolute false when $Mean = 0$ (respectively absolute true when $Mean = 1$) and singletons are not modified.

Next, the engine aggregates the decisions impact. Each tool has an initial impact which is modified given specific cases. Initial impact equals to 0 (without any impacts) or 1 (with the most impact) are not modified. Otherwise, at each n applicable case, the impact is altered with a weight (W, 25% if not valued in the CG) while kept bounded: $impact(n) = impact(n-1) - W \times impact(n-1)$ for a *lower* impact or $impact(n) = impact(n-1) + W \times (1 - impact(n-1))$ for a *greater* impact . Thus smaller steps are made for already extreme values (e.g. keeping very effective adaptations 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 limit of the total amount of impact usable. Thus this last classification is a knapsack problem. The available impact is the initial user impact total (a first step into profiling the user) minus the active decisions impact.

7 CASE STUDY

7.1 Scenario 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. We use the fly-over interaction technique (Boudoin et al., 2008). It is an implicit adaptive technique which offers a continuous interaction by detecting automatically the current task based on the cursor position. We use it as our 3D interaction base conjunctively to the engine. Besides, the task information is retrieved from fly-over which thus becomes a task sensor for the engine. The application tools are presented in Fig 11. The engine uses general rules as to:

1. Define possible interests:
   a) inside of a Zones Of Interest (ZOI);

   b) a previous interest in history.
2. Define possible objectives :
   a) selection of an object.
   b) navigation towards an object.
3. Define the will to enhance interests:
   a) a general will for all interests (Fig. 7).
   b) exceptions when an interest is part of a accomplished objective (e.g object selected).
4. Define possible enhancements:
   a) object visual modifications: color change.
   b) interaction modifications: visual/haptic force.
5. Define possible objectives assistances :
   a) select an object for a selection objective.
   b) Move toward an object for a selection or a navigation objective.
6. Manage adaptations states:
   a) remove visual modification for past interests;
   b) remove a currently applied force if the movement is abnormal (e.g local+high=the user is ”stuck”).
7. Manage decisions impacts:
   a) increase impact for some concepts: haptic impact> visual impact; interaction modification impact > visual modification impact;
   b) increase decision’s impact if present in history (Fig. 10);
   c) decrease interaction modification’s impact for local movement.
8. Define engine meta-adaptation:
   a) switch between engine configurations on the type of request (manual if the decision request came from the user)
   b) increase the total impact and decrease the decisions threshold for manual configuration.
   c) decrease the total impact and increase the decisions threshold for automatic configuration.

Finally, the application specific rules are:

9. Monitor the hand movement;
10. Ask for detected gesture attributes ;
11. Activate or deactivate a ZOI around the hand if a circular gesture occurred;
12. Activate a ZOI in the direction of the gesture if a rectilinear gesture occurred;
13. Deactivate this direction ZOI after 3s.
14. Deactivate every adaptations if the ”Z” gesture is detected.

7.2 Discussion

As a result, the rules combine themselves as expected (adaptations examples Fig. 12), but with supplementary outcomes which were not fully planned. Those results depend on the initial impact, confidence and total admissible impact values.

Type	Name	Aims
Sensor	ZOISens	Add and report the content of 3D zones
	PtFocus	Report the current objects beneath the pointer as interests
	SelSens	Report selected objects
	FOTaskSens	Report the current task
	GestureSens	Send recognized gestures name and attributes
	MvtSens	Qualify movement speed as high or low and movement scope as local or global
	AutoSens	Report if the decision is requested by the user or automatically
Actuator	ColorActu	Change the color of an object
	ForceActu	Add a haptic or a visual force to an object
	ZoomActu	Center the camera/zoom on an object
	SelActu	Add an object to the active selection
Meta-Actuator	ConfUser	Modify user attributes (e.g the total impact available, the confidence on impact ratio threshold etc.)

Figure 11: Applications available tools.

Interests and Objective Detection. In fact, there are several interest types: explicit, by creating voluntarily a ZOI (rules 1.a, 11, 12) or by centering the pointer on an object (PtFocus sensor), or implicit, either by moving rectilinearly toward an object (thus creating a ZOI: rules 1.a, 12) or by considering previous interest (rule 1.b). Confidence fusion of those information leads to the detection of the major interests, which should be the ones enhanced (rules 3). Objectives are specific recognized activities usually implicating an interest in a defined situation (rules 2). Objectives confidence depends on interests confidence but they have their own adaptations (rules 5).

Reactive Adaptations: Interest Enhancements

A Single Interest: Movement Influence. With an activated ZOI around the hand (rule 11), passing by

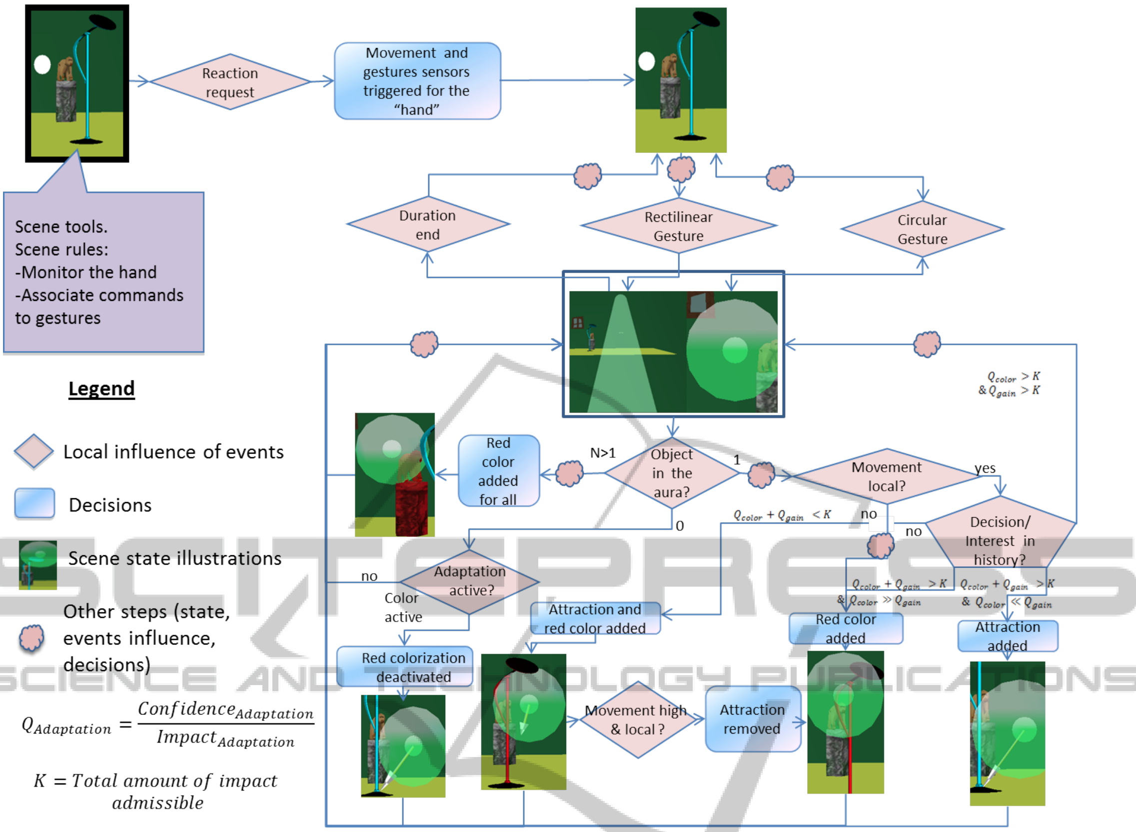


Figure 12: Illustration of some automatic adaptations depending on the context.

an object colors it red (rules 4.a, 7.a), while standing next to it makes it also attractive (as movement is then local, diminishing the attraction impact rules 4.b, 7.c). Colors are reset when the user moves far away while attraction is removed when the user tries to resist it (rules 6). When pointing an object or when moving toward an object during a global movement (rule 12), the object is colored red (rules 4.a, 7.a). When pointing an object from a rest position or starting a new movement directly toward an object (rule 12), it makes it also attractive (rules 4.b, 7.c: not intended at first, this primary intention can be highlighted due to the latency of the movement scope sensor, which still points the movement as local). Colors are reset after this ZOI deactivation (rule 13).

History Influence. In both previous cases, when pointed several times as an interest (thus present as several current facts or history facts, rule 1.b) attraction can be activated regardless of the movement scope (global or local). When it has been deactivated, attraction usually cannot be reactivated for a time corresponding to history memory (rule 7.b). Some reactivations occur for coloring as the decision has initially less impact. Moreover 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). Indeed an object can be simultaneously interesting enough to trigger the coloring adaptation (rule 4.a) and not enough to avoid the color reset (rule 6.a). In result, if an object has at a time a sufficient interest and then seems abandoned by the user, the object flickers for a while before its interest confidence reaches lower level. This is also a (unplanned) mean to attract the user attention.

Several Interests. More complex situations occur when several objects are close to the hand: e.g only adaptation with the less impact (coloring, rule 4.a) 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 admissible impact thus a group logic emerged). Situations are very various when several interests are spotted with more different confidences.

Pro-active Adaptations: Tasks Shortcuts. With the current task known (e.g selection or navigation), objects of interest can be interpreted as objectives (respectively to select an object or to navigate toward it, rules 2). Then if the interest confidence is high enough, a zoom on an object can occur (rule 5.b). In the case of the selection task, the object automatic selection can happen (rule 5.a). Those adaptations are pro-active as parts of the supposed intended task is actually done for the user. But those decisions are scarce since a lot of rules are involved. Thus the remaining confidence is usually low whereas the decisions impact is relatively high. Finally, a selected object is no longer the target of enhancement (as we can assume that the user is then well aware of it or has achieved this sub-objective, rule 3.b)

Meta-adaptations. The easiest way to obtain the previous pro-active adaptations is to switch to the manual mode. This reduces the engine requirement for adaptation through the meta-actuators (rules 8). Indeed if a user is the source of the adaptation request, the assistance need and a possible objective are very likely. With lower requirement in this context, a medium confidence is enough to obtain those assistances (a fulfilment of parts of the task, rules 5) .

Scenario Conclusion. Using the engine, we have successfully obtained an adaptations manager. The engine context and reasoning parts can also be grouped by their current role for the adaptations (Figure 13). The situation progresses with two interlaced processes: decision and comprehension. The decision process goes from representing the situation to reasoning, and transforms simple facts confidence into the best reaction bet. The comprehension process goes from identifying the situation to understanding how to assist (thus managing the two context-awareness trends) and transforms simple data to a potential full plan of the situation. The decision set is extended as the comprehension progresses. Based on the current events and knowledge, more information can be requested. Then adaptations are unlocked as the situation identification progresses. Reactive adaptations are available when an interest is acquired (e.g. the enhancement reactions). While still progressing

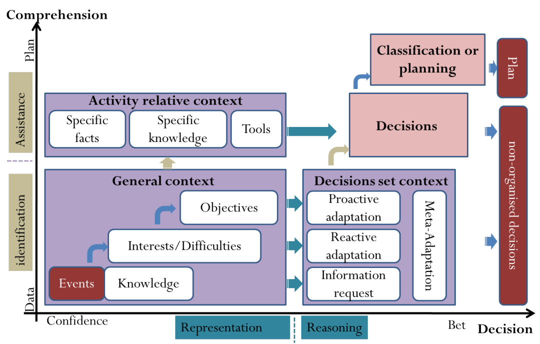


Figure 13: The adaptations manager realized with the engine (planning is not available yet).

through our reasoning and comprehension (e.g. fusing the interest information), objectives can be deducted and pro-active reactions obtained (e.g. the selection of a specific object). Meta-adaptations can be applied at each level (e.g. ease the reactions requirement based on the automatic mode information). All those reactions choices, parametrisation and reification depend on the current activity context. By having a semantic engine, it is easy to interfere on each concept. For example, a tool to retrieve the interests and their confidence from the engine can be used in order to benefit directly from the identification part. And the user can specify his objective to the engine and benefit directly from the assistance (without using automatic interest sensors to identify his intention).

8 CONCLUSIONS

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 lessened 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 the currently used ”impact” and rethinking 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 other data (than the 3D position) in order to learn and classify interesting situations.

Moreover, we can also enhance the task detection based on more context information than the fly-over position information. In fact this is a reinterpretation of this technique by splitting it in two and by inserting the engine in the middle. We use the sensor part which deduces the task based on the cursors position (within the control model) and the actuator part which is the selection of the control model (based on the task). Nevertheless, we will keep fly-over fully functional and allow the engine to force the control model through a specific actuator. In this case we benefit from both side: the good response time of fly-over and the ability to detect and manage more complex task recognition situations when needed.

Our next steps are: to continue to explore context, to make better tools descriptions and to allow more combinations (thus to further exploit the available context) and finally to proceed to engine evaluations.

REFERENCES

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). *Pervasive and Mobile Computing*.

Bonis, B., Stamos, J., Vosinakis, S., Andreou, I., and Panayiotopoulos, T. (2008). A platform for virtual museums with personalized content. *Multimedia Tools and Applications*, 42(2):139–159.

Boudoin, P., Otmane, S., and Mallem, M. (2008). Fly Over , a 3D Interaction Technique for Navigation in Virtual Environments Independent from Tracking Devices. *Virtual Reality*, (Vric).

Bouyer, G., Bourdot, P., and Ammi, M. (2007). Supervision of Task-Oriented Multimodal Rendering for VR Applications. *Communications*.

Bowman, D. A., Chen, J., Wingrave, C. A., Lucas, J., Ray, A., Polys, N. F., Li, Q., Haciahmetoglu, Y., Kim, J.-s., Kim, S., Boehringer, R., and Ni, T. (2006). New Directions in 3D User Interfaces. *International Journal*, 5.

Brézillon, P. (2011). From expert systems to context-based intelligent assistant systems : a testimony. *Engineering*, 26:19–24.

Celentano, A. and Nodari, M. (2004). Adaptive interaction in Web3D virtual worlds. *Proceedings of the ninth international conference on 3D Web technology - Web3D '04*, 1(212):41.

Chein, M. and Mugnier, M. (2009). *Graph-bases Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer.

Coppola, P., Mea, V. D., Gaspero, L. D., Lomuscio, R., Mischis, D., Mizzaro, S., Nazzi, E., Scagnetto, I., and Vassena, L. (2009). *AI Techniques in a Context-Aware Ubiquitous Environment*, pages 150–180.

Dey, A. and Abowd, G. (2000). 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.

Dey, A., Abowd, G., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *HumanComputer Interaction*, pages 1–67.

Frees, S. (2010). Context-driven interaction in immersive virtual environments. *Virtual Reality*, Volume 14(4):1–14.

Gu, T., Wang, X., Pung, H., and Zhang, D. (2004). An ontology-based context model in intelligent environments. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, volume 2004. Citeseer.

Irawati, S., Calderón, D., and Ko, H. (2005). 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.

Kabbaj, A. (2006). Development of Intelligent Systems and Multi-Agents Systems with Amine Platform. *Intelligence*, pages 1–14.

Latoschik, M. E., Biermann, P., and Wachsmuth, I. (2005). Knowledge in the Loop : Semantics Representation for Multimodal Simulative Environments (2005). pages 25 – 39.

Latoschik, M. E., Blach, R., and Iao, F. (2008). Semantic Modelling for Virtual Worlds A Novel Paradigm for Realtime Interactive Systems ? In *ACM symposium on Virtual reality software and technology*.

Lee, S., Lee, Y., Jang, S., and Woo, W. (2004). vr-UCAM: Unified context-aware application module for virtual reality. *Conference on Artificial Reality*.

Lugrin, J.-l. and Cavazza, M. (2007). Making Sense of Virtual Environments : Action Representation , Grounding and Common Sense. In *12th International conference on intelligent user interfaces*.

Octavia, J., Coninx, K., and Raymaekers, C. (2010). Enhancing User Interaction in Virtual Environments through Adaptive Personalized 3D Interaction Techniques. In *UMAP*.

Otterlo, M. (2009). *The logic of adaptive behavior*.

Peters, S. and Shrobe, H. E. (2003). Using Semantic Networks for Knowledge Representation in an Intelligent Environment. In *1st International Conference on Pervasive Computing and Communications*.

Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996). 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.

Ranganathan, A. and Campbell, R. H. (2003). An infrastructure for context-awareness based on first order logic. *Personal and Ubiquitous Computing*, 7(6):353–364.

Sowa, J. F. (2008). *Conceptual Graphs*, pages 213–217. Elsevier.

Wingrave, C. A., Bowman, D. A., and Ramakrishnan, N. (2002). Towards Preferences in Virtual Environment Interfaces. *Interfaces*.