A Semantics Driven User Interface for Virtual Saarlouis

Deborah Richards[1] and Stefan Warwas[2]

[1] Department of Computing Faculty of Science,
Macquarie University Sydney 2109, Australia
deborah.richards@mq.edu.au
[2] German Research Center for Artificial Intelligence,
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
stefan.warwas@dfki.de

Abstract. Virtual reality can support edutainment applications which seek to provide an engaging experience with virtual objects and spaces. However, these environments often contain scripted avatars and activities that lack the ability to respond or adapt to the user or situation; intelligent agent technology and a semantically annotated 3D world can address these problems. This paper shows how the use of agents and semantic annotations in the ISReal platform can be applied to the virtual heritage application involving the historic fortified town of Saarlouis.

Keywords: Semantic Virtual Worlds, Agents, Virtual Saarlouis, ISReal.

1 Introduction

While virtual reality is often associated with fantasy and entertainment, the application of virtual reality technology to travel and heritage applications for (re)creating actual and/or historical events and places is clear. Beside photo realistic graphics running in real-time, intelligent and flexible behavior of avatars and non player characters (NPCs) are required to produce a highly realistic virtual world. In the case of massively multi-player online role-playing games like World of Warcraft, avatar intelligence is primarily achieved by using real (i.e. human) intelligences who create their own (shared) meaning. System (artificial) intelligence is usually limited to the use of production-rule type triggers or hard-coded scripts to create the illusion of intelligent behavior in NPCs. The intelligent agent paradigm offers a clean, intuitive, and powerful way of modeling the behavior of intelligent entities in virtual worlds which tend to be dynamic, partially observable and non-deterministic. To address these challenges, agents can be represented in virtual worlds through avatars (their virtual bodies), interact with their environment through sensors and actuators, reason about their beliefs and plan their actions in order to achieve a given goal. In order to enable agents to interact with their virtual environment they need, beside purely geometric data, additional semantic information about their environment.

P. Anthony, M. Ishizuka, and D. Lukose (Eds.): PRICAI 2012, LNAI 7458, pp. 492–503, 2012.

The ISReal platform offers such features by combining technologies from various research areas such as computer graphics and the semantic web in a unique way [12]. In particular, highly realistic 3D simulation of scene objects are semantically annotated in a way that allows agents to reason upon them for their deliberative action planning in the semantic virtual world. Novelly semantics are not used purely to allow the software agent to reason; in the application reported in this paper involving the historic German town of Saarlouis, the object and environment semantics are used to allow the human user to intelligently navigate around and interact with the virtual world.

This paper first introduces the application domain of Saarlouis in Section 2 followed by consideration of the tourism and museum guide domain and approaches in Section 3. We then provide an overview of the ISReal platform in Section 4 looking in more detail at the agent component in Section 5. A semantics-driven user interface to Saarlouis is presented in Section 6. Section 7 concludes the paper and considers some future work.

2 Introducing Historical Saarlouis

The historical fortified town of Saarlouis was founded by Louis XIV in 1680 and built according to plans drawn up by the renowned fortress architect Vauban. The virtual environments we have created concern two distinct time periods for Saarlouis: 1700 and and 1870 covering the French (1680-1815) and Prussian (1815-1918) periods, respectively. Such time transitions are not possible in the real world, however, virtual worlds are not restricted by time or space.

Numerous avatars in period costume walk the streets of Saarlouis, including soliders and craftmen such as wigmakers, bakers and butchers. These characters have been created using the Xaitment[1] game engine and programmed to avoid collisions, detect the location of the user, to remain within defined walking areas and to move between specified locations. To act as intelligent agents, we have some specific semantically-annotated characters, e.g.Ludwig (Louis) XIV, founder and namesake of Saarlouis; Sebastian le Prestre de Vauban, city and fortress architect; Thomas de Choisy, the first governer and fortress engineer; Kiefer a wig maker; and Lacroix, the *forgotten soldier*. The annotated buildings or historical sites may be connected with the above people such as the Kommandantur, Lock Bridge, Vauban Island and Kaserne 1 Barracks. As with *people* objects, these *places* can provide services such as being open.

In addition to the user being able to meet a person or visit a place, the user is able to choose an *activity* (for example, the baker baking bread or the wigmaker making a wig). A particularly interesting activity in this application, is the fortification and protection of Saarlouis which is a complex process requiring many services that would need to be planned and multiple agents which would need to be coordinated. Here we only consider a simplified single agent version. We explore the fortification of Saarlouis further in Sections 4, 5 and 6.

[1] `http://www.xaitment.com/`

3 Virtual Guides for Objects of Interest

To assist us in the design of a user interface for Saarlouis, we considered applications and approaches to museum and tour guides as well as product and exhibit guides. All of these applications are concerned with presentation of objects of interest, rather than provision of information, to the user. The object of interest may be many different things such as a building, work of art, artefact, person, event, activity (e.g. cultural activity such as a folk dance). The object of interest may exist physically, conceptually, historically and/or virtually. The type of interaction and the devices used will depend on the nature of the object and whether the object and/or user are physically and/or virtually present in the environment. We are mostly interested in the situation where the user is telepresent. Some applications support both situations (e.g. the MINERVA [18] robot at the Smithsonian's National Museum of American History). We are particularly interested in smart applications which use Semantic Web and/or agent-based technologies.

The use of semantic annotations to enrich museum content has been considered within the PEACH project ([15], [16]). The Multimedia Data Base (MMDB) stores semantic annotation of audio, texts and images to enable multiple applications to browse the content repository via XML transactions. The annotations are "shallow representations of its meaning" where "*meaning* refers to a set of key words that represent the topics of each paragraph and image detail and that are organized in a classification." (page 4). The topics are keywords which represent entities (characters or animals) or processes (e.g. hunting or leisure activities) and can be assigned to a class using the *member-of* relation. By annotating the multimedia content, [16] are able to determine what story beats [10] should be provided and to support a number of "communicative strategies". One strategy proposes further links to other texts based on links in the current text. A second strategy uses the theme(s)/class(es) of the current text to suggest other links with the same theme or class. A third strategy offers specific examples related to the current text. We use a similar semantics-driven strategy (see Section 6), but navigation of a multimedia library and virtual Saarlouis are different experiences. A semantically annotated virtual world might be relevant in geographical simulation work. For example, the Crowdmags: Multi-Agent Geo-Simulation to manage the interactions of crowds and control forces [11], could semantically enhance the virtual environment (VE) to guide the agents in their decision making.

Kleinermann et al. [9] use stored navigation paths and semantics to reduce time and provide guidance on how to interact with an object in a VE. Kleinermann et al. seeks to address the difficulty of acquiring annotations. To maximise reuse of these annotations they have a tool to allow easy creation and modification of multiple sets of annotations to support different domains, tasks, purposes or groups of people. To address this issue we present a tool to assist with semantic annotation of our agents (see Section 5.3). However, our key focus is on how the annotations can be machine-processed by agents to guide user interactions (see Section 6).

4 An Overview of ISReal

This section provides an overview of the ISReal platform and its components [12]. The ISReal platform consists of components from different technology domains required for intelligent 3D simulation (see Figure 1).

Graphics. The central components regarding the computer graphics domain are XML3D[2] for managing the scene graph at run-time and the rendering engine which is responsible for visualizing the scene graph. The graphics components offer 3D animation services and virtual 3D sensors which can be used by agents to sense semantic objects. Furthermore, the platform can be extended by a virtual reality (VR) system for immersive user interaction in a given 3D scene.

Semantics and Services. The *Global Semantic Environment* (GSE) of the ISReal platform manages a global domain ontology describing the semantics of 3D objects, and a repository of semantic services for 3D object animations executed by the graphics environment. The ISReal platform consists of two semantic management systems (i) one for maintaining and querying the global ontology, and (ii) one for semantic service registration and execution handling. The ontology management system (OMS) allows to plugin different RDF triple stores of which one is uniquely selected for object queries in SPARQL, and different semantic reasoners like Pellet[3] (with internal Jena) and STAR[7] for complementary concept and relational object queries. The implementation uses the LarKC architecture [4]. The OMS is implemented as a LarKC Decider consisting of different LarKC reasoner plug-ins. As processing plug-ins the OWLIM triple store system [8] and Pellet are used.

Semantic World Model. The basic semantic grounding of the ISReal platform is called *semantic world model*. It is the set of semantically annotated scene graph objects with references to the global domain and task ontology, services, and hybrid automata. The semantics of each XML3D scene object is described by its RDFa[4] linkage to (i) an appropriate ontological concept and object definition in OWL, (ii) animation services in OWL-S, and (iii) hybrid automata for its behavior over continuous time.

Other Components. *Agent technology* is used to model the behavior of intelligent entities in a 3D scene. Due to the important role played in supporting user interactions and queries, Section 5 of this paper provides further information on the agent architecture and query handling for the ISReal platform. The *Verification* component manages and composes hybrid automata each describing temporal and spatial properties of 3D objects and their interaction in a scene, and verifies them against safety requirements given by the user at design time. The ISReal platform can run as a full fledged VR with immersive user interaction but it is also possible to connect to a scene with a web browser like Firefox or Chrome. User interaction involves a selection of choices and presentation of answers (in audio and text form) based on the semantics of the scene.

---

[2] `http://graphics.cs.uni-sb.de/489/`

[3] Pellet: `http://clarkparsia.com/pellet`

[4] `http://www.w3.org/TR/xhtml-rdfa-primer/`

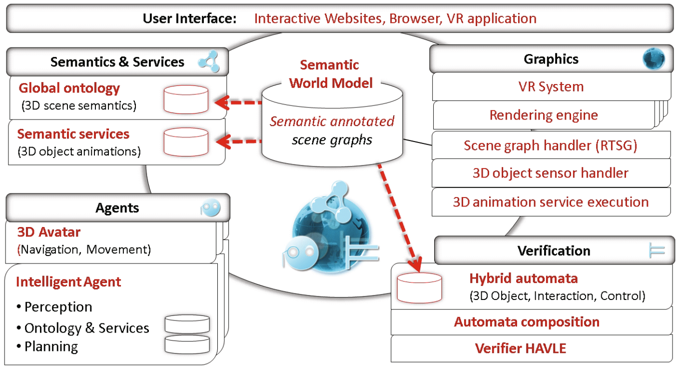

Fig. 1. The ISReal platform components

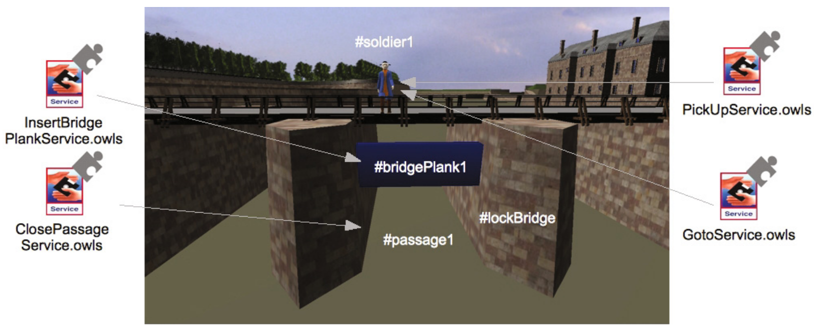

Fig. 2. Simulation of historic Saarlouis

Related Work. In different contexts, others have considered semantic annotation of 3D environments e.g. in [13]. Kalman et al. [6] proposed the concept of *smart objects* which is a geometrical object enriched with meta information about how to interact with it (e.g. grasp points). Abaci et al. [1] extended smart objects with PDDL data in order to plan with them. There are numerous examples of game engines being connected to BDI agents (e.g. [3]) and virtual environments using multi-agent systems technology (e.g. [2]). ISReal differs from this body of work as it integrates virtual worlds, semantic web and agent technology into one coherent platform for semantically-enabled 3D simulation of realities. Work using classical planners aim on improving the look-ahead behavior of BDI agents and on explicit invocation of a classical planner for sub-goals, assuming plan templates to be static at run-time. Our agents discover new services at run-time.

Saarlouis Context. As an example of how ISReal and Semantic Web technologies are used in the Saarlouis application, see Figure 2. The webservices shown allow a novice soldier agent to learn how the fortifications, particularly the lock bridge, can be used to protect Saarlouis from attack.

5 The ISReal Agent

Intelligent agents that inhabit virtual worlds require a flexible architecture in order to reason and plan in real-time with the information they perceive from their environment. Section 5.1 provides on overview of the general ISReal agent architecture. Section 5.2 introduces the ISReal query taxonomy. Finally Section 5.3 explains how to develop ISReal agents.

5.1 Architecture

ISReal agents are based on the *belief-desire-intention* (BDI) agent architecture which is well suited for dynamic and real-time environments [14]. The ISReal agent controls an avatar (its virtual body) which is situated in a 3D scene. We extended the agent with an individual sensor-based object perception facility (its interface to the virtual world), and a component for handling semantic web data, called *Local Semantic Environment* (LSE). In ISReal we follow the paradigm "*What you perceive is what you know*", meaning an agent only has knowledge about object instances and services it already perceived during run-time through its sensor(s) and the components that manage the global semantic world model (GSE and RTSG) are strictly passive. This paradigm enables us to simulate realistic information gathering. Figure 3 shows an overview of the ISReal agent architecture. The agent's BDI plan library implements (i) the basic interaction with the ISReal platform and its LSE (e.g. initialization and perception handling), (ii) the agent's domain independent behavior (e.g. basic movement) and (iii) domain specific behavior patterns (customized behaviors regarding a certain domain or application). The domain independent behavior patterns allow us to deploy an agent into a virtual world it is not explicitly designed for and it is still able to perform basic tasks.Section 6 provides an example.

The LocalSE contains the agent's knowledge base. It is important to note that the agent's local knowledge is probably incomplete and can be inconsistent to the real world (e.g. due to outdated knowledge). The LocalSE is only updated if the agent perceives something. The LocalSE consists of (i) an *Ontology Management System* (OMS) to a) govern the semantic description of the world that is known to the agent, b) allow queries in SPARQL[5], a query language to primarily query RDF graphs, and c) semantic reasoning, and (ii) a service registry to maintain semantic services for the interaction with the objects in the 3D environment (object services). In addition it also provides tools for semantic service handling like a *service composition planner* (SCP) and a matchmaker.

5.2 Query Handling

Users need an interface for querying their user agents and assigning goals to them. To achieve a given goal, agents need powerful reasoners to access the knowledge in their LocalSE. For this purpose we created a query taxonomy. We use a plug-in architecture for query types to be open for new technologies

[5] W3C: `http://www.w3.org/TR/rdf-sparql-query`

and tools. We distinguish between (i) informational queries for querying the agent's local knowledge base, and (ii) transactional queries that cause the agent to perform actions in order to reach an assigned goal. Query classes include:

Spatial Query. To determine the spatial relationship between objects. For example, *"Is object A inside object B?"*.

Object Query. To retrieve information about instances (A-Box knowledge), such as *"Is passage1 open?"*. To handle these queries (expressed in SPARQL[6]) we use the RDF triple store system OWLIM [8].

Concept Query. Concept queries, such as *"Is Passage a subclass of Open?"*, retrieve information about concepts (T-Box knowledge). We use the OWL2 semantic reasoner Pellet to answer such queries.

Object Relation Query. To find non-obvious relations between different objects, i.e. a set of entities $\{e_1, \ldots, e_n\}$. The OMS can find the smallest tree of the RDF graph representing the KB, such that it contains all entities $\{e_1, \ldots, e_n\}$ [7]. A simple query is *"How are objects passage1, passage2, passage3, and bridgeA related?"*.

Funcional Relational Query. To answer questions like *"How to close passage X?"*. The query is purely informational and does not cause the agent to execute the found plan.

Temporal Query such as *"When did you see object X the last time?"*.

Matchmaking Query. The user can ask the agent whether it knows a certain type of service. The query is answered by using a semantic matchmaker.

Declarative Goal Query. A declarative goal specifies a target state the agent should bring about (e.g. $at(agent1, bridgeA)$). The agent has to do means-end reasoning and execute the found plan (if there is one).

Perform Goal Query. Perform goals are purely procedural goals that do not contain declarative information about the goal state. Perform goals trigger agent plans that are able to achieve the desired behavior (e.g. turn around).

5.3 Developing and Semantically Annotating ISReal Agents

To provide a complete agent configuration (see Figure 4) a graphical designer has to develop the body geometry, movement animations, and position the sensor in XML3D, using a state of the art 3D modeling tool. From the agent perspective we assume that the domain ontology already exists, so that the avatar can be semantically annotated using our annotation tool. OWL-S service descriptions have to be developed and implemented that describe the agent's basic actions (OWLS-ED). The initial agent knowledge base (A-Box) can be created using an ontology tool like Protege[7]. The modeling of the actual agent and its behavior is done using our model-driven development environment for multiagent systems *DSML4MAS Development Environment* (DDE) [19].

[6] SPARQL: `http://www.w3.org/TR/rdf-sparql-query/`
[7] Protege: `http://protege.stanford.edu/`

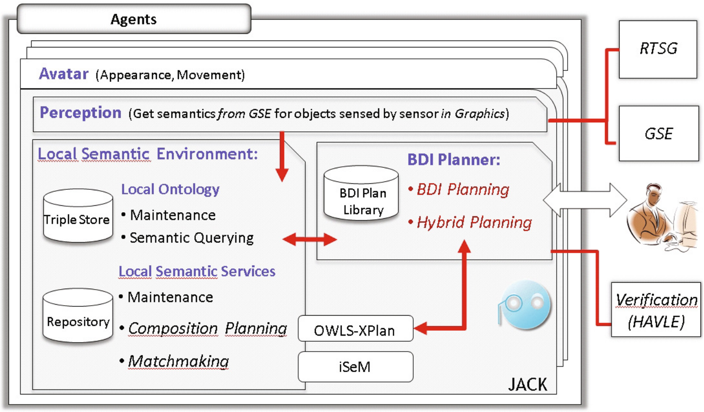

Fig. 3. ISReal agent architecture

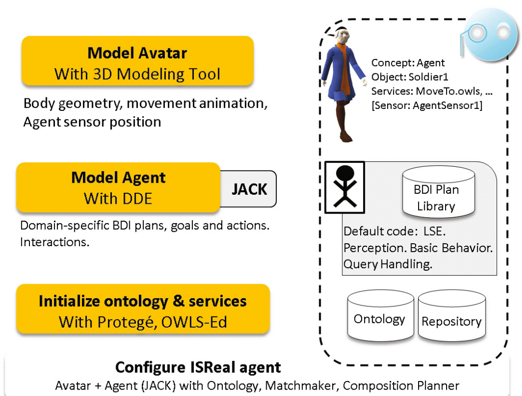

Fig. 4. ISReal agent configuration

6 Semantics and Agent-Driven User Interface

While the ISReal architecture supports a very large number of possible queries, rather than require the user to form the query or our system to interpret the request, we use the object semantics and agent reasoning and planning capabilities to determine and present what queries are available. Basically, in the approach the user is given the opportunity to select an object of interest and later to follow links related to that object (see how objects can be related to one another in Table 1), return to the original object or select a new object of interest. To create an engaging experience which will allow more of Saarlouis to be visited, satisfy user preferences and achieve teacher/curator learning goals, the ability of the agent to perceive the objects in the environment and to plan will be used to add elements of serendepity or dramatic effect "on the way" to the user's chosen object.

Table 1. Partial meta-model for three *Person* objects

Property	Example 1	Example 2	Example 3
Name:string	Ludwig XIV	Sebastian de Vauban	Thomas de Choisy
Date of birth:date	5-11-1638	??-5-1633	1632
Place of birth:Place	St Germain en Laye	St-Leger Moigneville	
Date of death: date	1-11-1715	30-3-1707	20-2-1710
Place of death: Place	Versailles	Paris	Saarlouis
Lived In:Place	Versailles		Kommandantur
Commissioned:Person	Vauban	Choisy	
Role:string	King of France	Architect of Saarlouis	The first governor
services		introduceSelf	introduceSelf explainFortifications

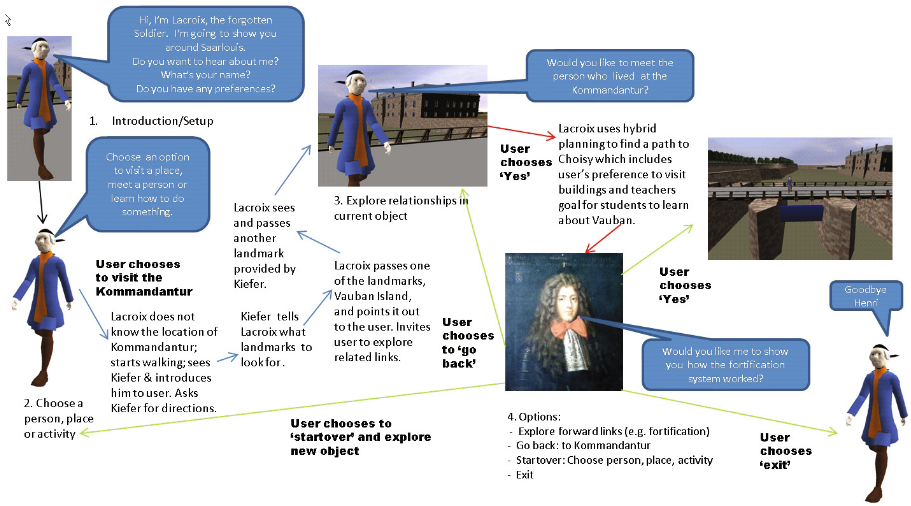

Fig. 5. Possible interaction between user, agent and environment

Example. A possible interaction is depicted in Figure 5 where we see one complete session starting from setup to exit. Not shown in the diagram, when the user starts up the virtual environment, the user (who may be a tourist, history buff or school child) is able to select their personal guide, consisting of the avatar and its role (here novice). The intelligent guide agent is attached to the avatar and initialized. As shown in step 1 the agent introduces themself and requests some information from the user (name and preferences).

As a basic interaction strategy, the user is asked (see step 2) to choose if they want to meet a person, visit a place or learn about an activity/task. Objects can be classified as either a place or person and they may know how to perform an activity (e.g. introduceSelf, explainFortifications). As this knowledge is part of the object meta-model, when the user selects a person, place or activity, the user-interface can query the objects in the world/scene to produce an appropriate list for the user to choose from. This list will take into account what the user has already seen, either hiding already visited objects from the list or placing them at the end of the list. User and teacher preferences could also be taken into account in the ordering provided. Once a user chooses the object of interest, the agent is assigned the goal to find an "interesting" pathway from the current location in the virtual world to the location of the chosen object.

Three simplified *tours* are depicted, each in a different shade (tour 1-blue, tour 2-red, tour 3-green). Four alternative green tours are shown and reflect the choices that were available to the user. The end node of each tour depicts the choice taken by the user. Alternative paths from one user-chosen end node to another are possible. Tours 1 and 2 show one possible path that might have occured. To determine the path from the current location to the object chosen by the user or to determine if an object passed along a pathway should be brought to the attention of the user, the agent uses a number of filters and strategies. These filters are based on the user's preferences for certain types of objects or themes; the existence of a pedagogical model which identifies any objects or activities which must be encountered (these are not necessarily added to the first tour) or the sequence in which certain objects must be visited; the possible inclusion of dramatic elements such as a visit to the house of a colourful character or building with mysterious past. The agents access to services and ability to reason and plan about the objects and environment allow factors such, as if the place has already been visited, to be taken into account. Adding further to the element of surprise, a strategy recommended by [16] to improve attention and memory retention that needs to be balanced with meeting user's expectations , and drawing on the ability of the agent to perceive its world to discover new objects and services, we also allow the agent to interrupt its current plan to, for instance, introduce a place, person or activity discovered on the way. See, for example, the disovery of Kiefer and Vauban Island in Tour 1(blue). The relationships between objects, and objects and services, are used to drive possible further interactions. Similar to the communicative strategies used in the PEACH project (Section 3), in step 3 the user is able to follow a link related to the Kommandantur, i.e. to visit Choisy who lived in this building (see Table 1). This allows the user to traverse between objects in a forward fashion. Note that other links and options were also possible for the user but not shown for clarity in the example: step 4 depicts a more complete set of options. As well as following links forward, the user can choose to **go back** to access options related to the previously chosen object, **start over** with a top level search for place, person or activity or **exit** to quit the program. At any point, the user is able to take these three options.

Query Processing. As introduced in Section 5.2, the agent is capable of handling several types of queries. These queries are primarily used by the agent to satisfy its own informational or transactional needs, rather than for the end user as we believe they would have some difficulty identifying which type of query was most appropriate and how to form the query. For the reasons given above, for the Saarlouis application we do not allow the user to enter natural language requests. For the Saarlouis application we may allow the user to select certain queries in order, for instance, to satisfy their own particular interests or a set of learning goals set by a teacher.

In the context of Saarlouis and in line with Silveira et al. [17] user information needs, our approach provides information which addresses: *Choice:* what places can I choose to visit?; who can I choose to meet?; *Procedural:* how do I do $< activity >$; where can I find $< person >$?; how do I get to $< place >$; in what order should I visit Saarlouis? *Informative:* what/who can I do/see/visit/meet? *Guidance:* who should I talk to next?; where should I visit next?; who should I ask? What should I do now? *History:* who have I already met?; what have I already visited?; what have I already done? *Descriptive:* what does this person do?; who is this?; why are they famous?; who lived here? what can you tell me about $< person, place, activity >$?; how is this person related to $< person, place >$?

Silvieira's user information needs which are not supported relate more to explanations and system motivations and include: *Interpretive:* why are you telling me about $< object >$?; where are we going?; why are we going to see $< object >$; *Investigative:* did I miss anything?; what happened before that time?; what do I still need to do?; and *Navigational:* where am I? In line with the findings of [5] there are some additional status-related questions that could be considered to allow greater transparency and minimize user frustration: What are you doing now? Why did you do that? When will you be finished? What sources did you use? The status, interpretive, investigative and navigational questions we intend to offer in an interrupt mode allowing the user at any time to pause the current explanation, choose the question of interest to them and then return to the current interaction. These questions may be affected by which time period was chosen and other user preferences.

In ISReal the user is able to select and right-click on an object to receive the set of services it offers and a description. These services can be selected and added to a stack for execution by the agent. We anticipate that in addition we will offer the user a set of relevant queries for that object based on the query-type modules available in ISReal and in line with the types of questions posed above. Being able to click on an object in the scene and gain information about it and execute its services provides an alternative and complementary interaction method and would better suit a user or application where being taken on a tour is less appropriate or appealing.

7 Conclusion

We have presented the use of the ISReal platform in the Saarlouis virtual heritage application. Rather than providing a virtual world for free exploration, our intelligent agent acts as a tour guide that escorts the user around Saarlouis, providing explanations and allowing choices at certain points in the tour. The agent uses the semantics embedded in the world to dynamically plan suitable tours that take into account user preferences and history, pedagogical goals and potentially incorporate dramatic elements to make the experience more enjoyable and memorable.

References

1. Abaci, T., Ciger, J.: Planning with Smart Objects. In: WSCG SHORT Papers Proceedings, pp. 25–28. UNION Agency - Science Press (2005)
2. Anastassakis, G., Ritchings, T., Panayiotopoulos, T.: Multi-agent Systems as Intelligent Virtual Environments. In: Baader, F., Brewka, G., Eiter, T. (eds.) KI 2001. LNCS (LNAI), vol. 2174, pp. 381–395. Springer, Heidelberg (2001)
3. Davies, N.P., Mehdi, Q.: BDI for Intelligent Agents in Computer Games. In: Proc. 8th Int. Conf. on Comp. Games, CGAMES 2006. Uni. of Wolverhampton (2006)
4. Fensel, D., et al.: Towards LarKC: a Platform for Web-scale Reasoning. IEEE Computer Society Press, Los Alamitos (2008)
5. Glass, A., McGuinness, D.L., Wolverton, M.: Toward establishing trust in adaptive agents. In: Proc. 13th IUI 2008, Gran Canaria, Spain, pp. 227–236. ACM, NY (2008)
6. Kallmann, M.: Object Interaction in Real-Time Virtual Environments. PhD thesis, École Polytechnique Fédérale de Lausanne (2001)
7. Kasneci, G., Ramanath, M., Sozio, M., Suchanek, F.M., Weikum: STAR: Steiner Tree Approximation in Relationship-Graphs. In: Proc. of 25th IEEE Intl. Conf. on Data Engineering, ICDE (2009)
8. Kiryakov, A., Ognyanov, D., Manov, D.: OWLIM - a Pragmatic Semantic Repository for OWL. In: WISE Workshops, pp. 182-192 (2005)
9. Kleinermann, F., De Troyer, O., Creelle, C., Pellens, B.: Adding Semantic Annotations, Navigation Paths and Tour Guides to Existing Virtual Environments. In: Wyeld, T.G., Kenderdine, S., Docherty, M. (eds.) VSMM 2007. LNCS, vol. 4820, pp. 100–111. Springer, Heidelberg (2008)
10. Mateas, M., Stern, A.: Towards Integrating Plot and Character for Interactive Drama. In: Socially Intelligent Agents, pp. 221–228 (2002)
11. Moulin, B., Larochelle, B.: Crowdmags: Multi-Agent Geo-Simulation of Crowd and Control Forces Interactions. In: Mohamed, A. (ed.) Modelling, Simulation and Identification. InTech (2010), `http://www.intechopen.com/books/modelling--simulation-and-identification/title-crowdmags-multi-agent-geo-simulation-of-crowd-and-control-forces-interactions-`
12. Nesbigall, S., Warwas, S., Kapahnke, P., Schubotz, R., Klusch, M., Fischer, K., Slusallek, P.: Intelligent Agents for Semantic Simulated Realities - The ISReal Platform. In: Proc. 2nd Conf. on Agents & AI, ICAART 2010, pp. 72–79 (2010)
13. Pittarello, F., De Faveri, A.: Semantic Description of 3D Environments: A Proposal Based on Web Standarts. In: Proc. of Intl. Web3D Conf. ACM Press (2006)
14. Rao, A.S., Georgeff, M.P.: BDI Agents: From Theory to Practice. In: Proc. of the 1st Intl. Conf. on Multi-Agent Systems, pp. 312–319. AAAI Press (1995)
15. Rocchi, C., Stock, O., Zancanaro, M.: Semantic-based Multimedia Representations for the Museum Experience. In: Proc. HCI in Mobile Guides Wkshp, Mobile HCI 2003 (2003)
16. Rocchi, C., Stock, O., Zancanaro, M., Kruppa, M., Krueger, A.: The Museum Visit: Generating Seamless Personalized Presentations on Multiple Devices. In: Proc. Intelligent User Interfaces IUI 2004, pp. 316–318. ACM Press (2004)
17. Silveira, M., de Souza, C., Barbosa, S.: Semiotic Engineering Contributions for Designing Online Help Systems. In: SIGDOC 2001. ACM (2001)
18. Thrun, S., et al.: MINERVA: A second generation mobile tour-guide robot. In: Proc. Int. Conf. on Robotics & Automation, ICRA 1999, Detroit, Mich, May 10-15 (1999)
19. Warwas, S., et al.: The DSML4MAS Development Environment. In: Proc. of 8th Intl. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2009 (2009)