ISReal: An Open Platform for Semantic-Based 3D Simulations in the 3D Internet

Patrick Kapahnke, Pascal Liedtke, Stefan Nesbigall,
Stefan Warwas, and Matthias Klusch *

German Research Center for Artificial Intelligence, Saarbrücken, Germany
`firstname.surname@dfki.de`

Abstract. We present the first open and cross-disciplinary 3D Internet research platform, called ISReal, for intelligent 3D simulation of realities. Its core innovation is the comprehensively integrated application of semantic Web technologies, semantic services, intelligent agents, verification and 3D graphics for this purpose. In this paper, we focus on the interplay between its components for semantic XML3D scene query processing and semantic 3D animation service handling, as well as the semantic-based perception and action planning with coupled semantic service composition by agent-controlled avatars in a virtual world. We demonstrate the use of the implemented platform for semantic-based 3D simulations in a small virtual world example with an intelligent user avatar and discuss results of the platform performance evaluation.

1 Introduction

In the Internet of today, navigation and display of content mostly remains two-dimensional. On the other hand, the proliferation of advanced 3D graphics for multi-player online games, affordable networked high-definition display and augmented reality devices let Internet users increasingly become accustomed to and expect high-quality 3D imagery and immersive online experience. The 3D Internet (3DI) is the set of 3D virtual and mixed reality worlds in the Internet that users can immersively experience, use and share with others for various applications [25, 5, 2, 31].

As of today, the 3DI offers, for example, various alternative worlds like SecondLife (2L)[1], questville, Croquet and WorldOfWarcraft, and mirror worlds like Twinity[2]. Applications include socializing and business collaboration in 3D meeting spaces, the 3D exploration of virtual cities, the participation in cross-media edutainment events like concerts and lectures, the trading of real and virtual assets, the functional 3D simulation of production lines and architecture at design time, as well as advanced visual 3D information search by using 3D Web

---

* The work presented in this paper has been partially funded by the German Ministry for Education and Research (BMB+F) under project grant 01IWO8005 (ISReal).

[1] http://secondlife.com

[2] http://www.twinity.com

browsers such as SpaceTime and ExitReality. In such virtual worlds, the user is usually represented by and driving the behavior of an avatar as her digital alter-ego.

Major challenges of the 3DI are (a) the more realistic, standard-based 3D graphical display in 3D Web browsers, and (b) the making of user avatars behave more intelligent in their 3D environment. For example, the intelligence of most avatars in virtual worlds today is either restricted to direct execution of non-verbal user commands, or rather simple event-rule-based but resource-optimized means of AI planning with massive volumes of action scripts in online games. Besides, in most cases, avatars are not even capable of understanding the semantics of their perceived 3D environment due to the lack of standard-based semantic annotations of 3D scenes and reasoning upon them or do not exploit 3D scene semantics for intelligent action planning in a virtual world they are involved in.

To address these challenges, we developed the first open, cross-disciplinary 3DI research platform, called ISReal, that integrates semantic Web, semantic services, agents and 3D graphics for intelligent 3D simulation of realities. In this paper, we describe the innovative interplay between its components with focus on semantic 3D scene annotation and query processing, and the semantic-based action planning of intelligent agent-controlled avatars together with a discussion of our experimental performance evaluation of the platform in a simple virtual 3D world. To the best of our knowledge, there is no other such integrated 3DI platform available yet.[3]

The remainder of the paper is structured as follows. Section 2 provides an overview of the ISReal platform while sections 3 and 4 describe the global semantics and intelligent agents for semantic-based 3D simulation. Section 5 demonstrates the use of the platform for a simple use case, followed by performance evaluation results and comments on related work in Sections 7 and 8.

2 ISReal Platform: Overview

Virtual world descriptions in XML3D. The ISReal platform can be used to develop and simulate virtual worlds in XML3D[4] which is a 3D graphics-oriented extension of HTML4. A virtual world scene is graphically described in form of a single XML3D scene graph that includes all objects of the 3D scene to be displayed as its nodes. In contrast to X3D[5], XML3D scene descriptions can be directly embedded into a standard HTML page such that every scene object becomes part of and accessible in the standard HTML-DOM (Document Object Model) by any XML3D-compliant Web browser capable of rendering the scene without any specific viewer plug-in required. Graphical changes in the virtual world during its simulation such as user interactions with the scene and 3D object

---

[3] Major barriers of an 3DI uptake by people today refer to its potential physio-cognitive, social and economic impacts on individual users of virtual worlds which discussion is outside the scope of this paper.

[4] http://www.xml3d.org

[5] http://www.web3d.org/x3d/specifications/

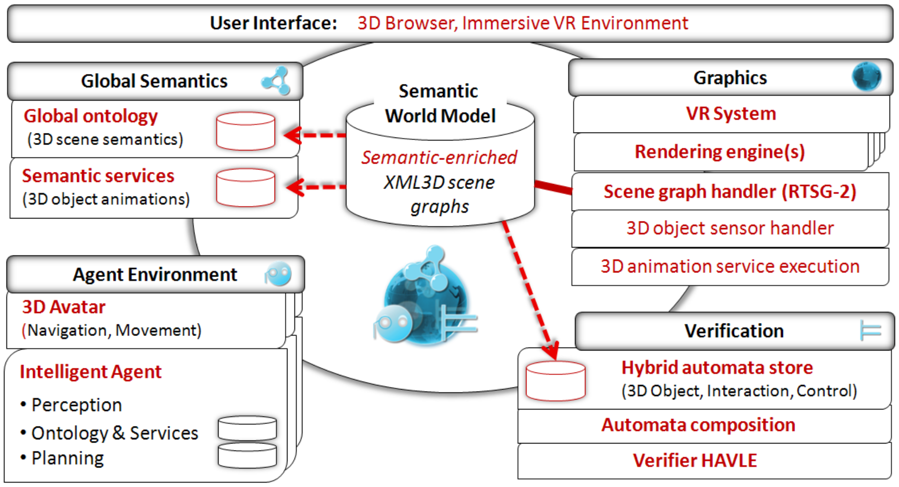


Fig. 1. ISReal platform components.

animation in the browser correspond to changes of its XML3D scene graph in the Web page of the virtual world scene which is loaded and processed by the ISReal client. In the following, we give an overview of the platform components and its communication architecture for virtual 3D world simulations.

Platform Components. The ISReal platform consists of five groups of components that are the user interface, the global semantics, 3D graphics, intelligent agents and verification environment (see Fig. 1). The graphics environment maintains the given set of XML3D scene graphs of virtual worlds by its internal RTSG-2 (real-time scene graph) system [24] and renders them by a pluged-in 3D rendering engine for high-quality 3D display such as our world-fastest raytracer RTFact [9] at run time. For immersive 3D interaction with simulated virtual worlds, it additionally provides an open, immersive VR (virtual reality) system. The global semantics environment (GSE) is responsible for managing global scene ontologies each of which describing the semantics of a virtual world in its application domain as well as the execution handling of globally registered semantic services which groundings have an effect on these ontologies such as the change of the position of some object in the scene by the respective 3D animation in the graphics environment (cf. Section 3). The verification environment manages and composes hybrid automata that describe spatial and temporal properties of scene objects and their interactions and verifies them against given safety requirements at design time; for reasons of space, we omit a description of this platform component. The semantic world model of the platform is the set of semantically annotated 3D scene graphs with references to the global semantics and the verification component. The agent environment manages the avatar-controlling intelligent agents capable of scene perception, local scene ontology management and semantic-based action planning to accomplish its tasks given

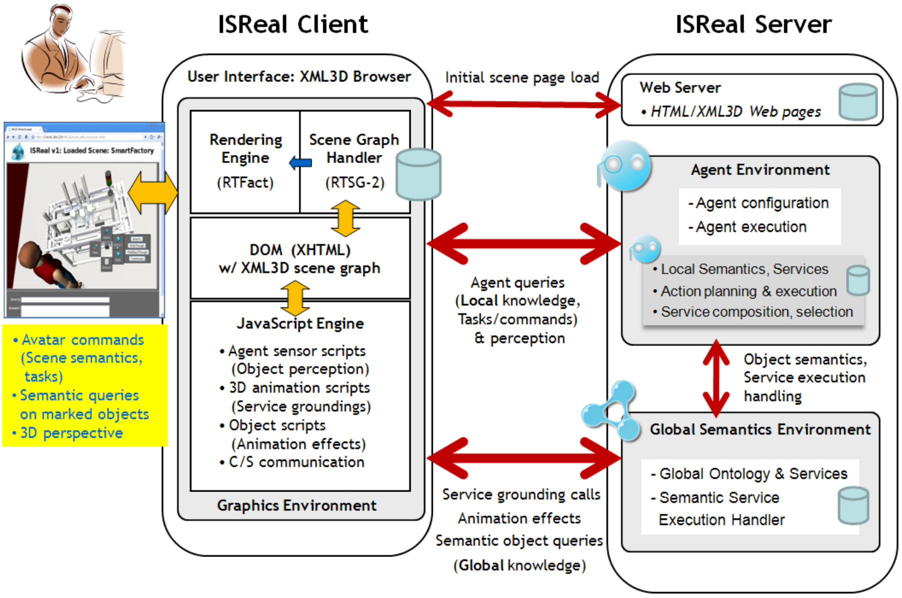


Fig. 2. ISReal v1.1 communication architecture for virtual 3D world simulations.

by the user or other agents (cf. Section 4). Finally, the user can interact with a 3D simulated virtual world scene by alternative means of 3D Web-based or immersive 3D virtual reality system-based user interface of the platform. The interface is either of both XML3D-compliant versions of Google Chrome and Mozilla Firefox browsers or an immersive VR environment based on the open VR system Lightning (which we connected with multi-touch display, space mouse, tracking system and iPhone as 3D input devices). A user can (non-verbally) query the semantics of marked single objects in the simulated scene with or without her avatar, and to command her avatar to answer complex semantic queries and to pursue given tasks in the scene.

Communication Architecture for Virtual 3D World Simulation. The client-server-based communication architecture of the ISReal platform 1.1 for single-user virtual 3D world simulation is shown in Figure 2.

The ISReal client is exclusively responsible for maintaining and rendering the complete virtual world scene with its embedded 3D graphics environment, and communicates with the ISReal server hosting all other components (and a Web server, in case of XML3D browser as ISReal client) for intelligent simulation. Asynchronous and bidirectional client-server communication is impemented by use of the WebSockets API[6]. Once the initial world scene page in HTML/XML3D is loaded by the ISReal client from the ISReal server, the client connects to server-sided components, triggers the scene-relevant configuration of semantics

---

[6] http://dev.w3.org/html5/websockets/

and agents at the server, and is responsible for user-interaction-based updates and rendering of the XML3D scene graph (Web-based or immersive 3D).[7] The open ISReal platform in its current version 1.1 has been fully implemented in Java and JavaScript.

3 ISReal Global Semantics

Semantic-based 3D simulation of virtual worlds is a key feature of the ISReal platform. In this section, we describe the semantic annotation of 3D scene objects and the global semantics environment of the platform in more detail.

3.1 Semantic 3D Scene Object Annotation

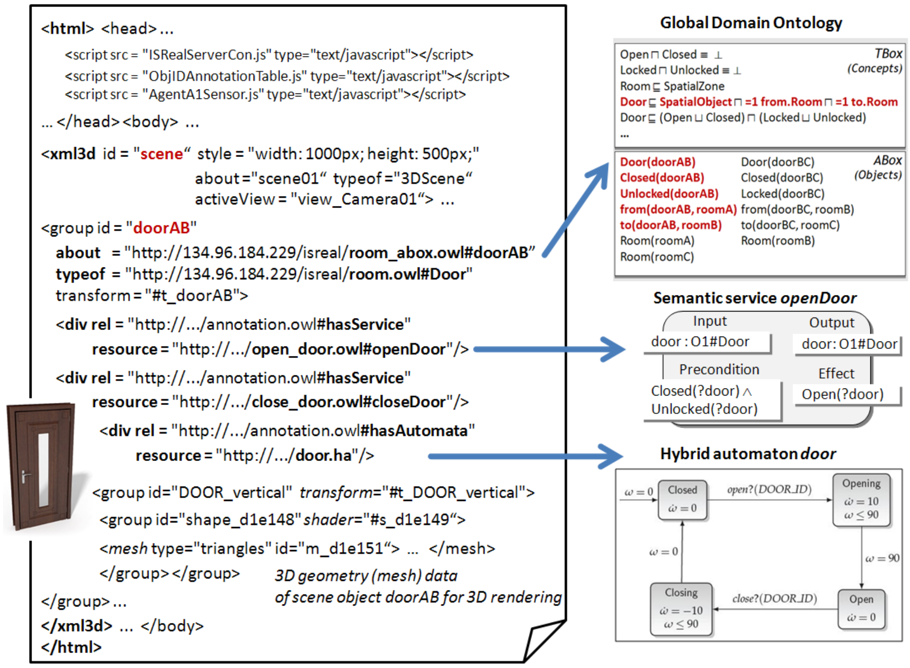
The semantic world model of the platform is the set of all semantically annotated XML3D scene graphs for simulated virtual worlds. Any 3D scene object in a virtual world is represented as a node of the XML3D scene graph that graphically describes this world. The semantics of a 3D scene object can be described by annotating its XML3D scene graph node by use of standard RDFa[8] with links to (a) the uniquely assigned object in a given global scene ontology described in (the OWL-Horst fragment of) standard OWL2 that represents the conceptual and assertional knowledge about the scene and application domain, (b) semantic services in OWL-S that describe the operational functionality of the scene object and are grounded in respective 3D animation scripts, and (c) hybrid automata that describe object properties with respect to continuous time and space in FOL linear arithmetics.

Figure 3 shows an example of semantic annotation of a virtual worlds scene object, that is a door connecting room A with room B. The representation of this object in the XML3D scene graph refers to a node labeled "doorAB" that includes its graphical description and semantic annotation. The first case refers to the 3D geometry (mesh) data required for rendering the scene object "doorAB" as defined in its respective subnode. The semantic annotation of the "doorAB" node is in RDFa with references to (a) an uniquely assigned object "doorAB" which semantics is defined in a given global scene ontology, (b) a set of semantic services describing the opening and closing of "doorAB" each of which grounded with an appropriate 3D animation script to be executed by the graphics environment, and (c) a hybrid automaton describing the temporal-spatial property that "doorAB" can be opened and closed with angular speed of 10 degrees per second, which is not possible to encode and reason upon in OWL2. Both the given global ontology and semantic object services are maintained in the global semantics environment of the ISReal platform.

---

[7] We are working on a multi-user/server architecture where the ISReal server maintains the global scene graph and provides multiple clients with only update instructions of how to change and render their local views on the scene based on user interaction events.

[8] The same principle of semantic annotation can be applied to X3D scene graphs as well. For a discussion of the benefits of XML3D over X3D, we refer to [27].



```
<html> <head>...
    <script src = "ISRealServerCon.js" type="text/javascript"></script>
    <script src = "ObjIDAnnotationTable.js" type="text/javascript"></script>
    <script src = "AgentA1Sensor.js" type="text/javascript"></script>
... </head> <body> ...
<xml3d id = "scene“ style = "width: 1000px; height: 500px;"
                    about ="scene01“ typeof ="3DScene“
                    activeView = "view_Camera01“> ...
<group id = "doorAB"
  about   = "http://134.96.184.229/isreal/room_abox.owl#doorAB”
  typeof  = "http://134.96.184.229/isreal/room.owl#Door"
  transform = "#t_doorAB">
  <div rel = "http://.../annotation.owl#hasService"
        resource = "http://.../open_door.owl#openDoor"/>
  <div rel = "http://.../annotation.owl#hasService"
        resource = "http://.../close_door.owl#closeDoor"/>
    <div rel = "http://.../annotation.owl#hasAutomata"
          resource = "http://.../door.ha"/>
        <group id="DOOR_vertical"  transform="#t_DOOR_vertical">
        <group id="shape_d1e148" shader="#s_d1e149“>
        <mesh type="triangles" id="m_d1e151“> ...  </mesh>
        </group> </group>
</group> ...
</xml3d> ... </body>
</html>
```

Fig. 3. Example of semantic annotation of a XML3D scene graph object with RDFa.

3.2 Global Semantics Environment

Architecture. The global semantics environment (GSE) consists of two components as shown in Figure 4, that are the global ontology management system (OMS) and the semantic service handler (SemSH). The OMS maintains a given set of global ontologies each of which describing the conceptual (TBox) and factual (ABox, fact base) knowledge about one simulated virtual world in OWL2. It handles the processing of different types of semantic queries issued by the user or agents against the global ontology of the actually simulated virtual world[9]. We assume that the TBox of the global ontology, in contrast to its ABox, does not change during simulation. The selected global scene ontology is materialized in, updated and queried through a selected RDF store of the OMS as usual. Other semantic queries (which answering is not possible by triple stores) are routed by the OMS query decider to the appropriate semantic reasoner(s) depending on its type or indicated by the user. The SemSH maintains the global semantic service repository that is assumed to contain all services in OWL-S which are related to the global scene ontology in terms of having either a precondition to be checked against its fact base, a grounding that may update the fact base as an effect, or both.

---

[9] In the following, we focus on the global ontology of one virtual world.

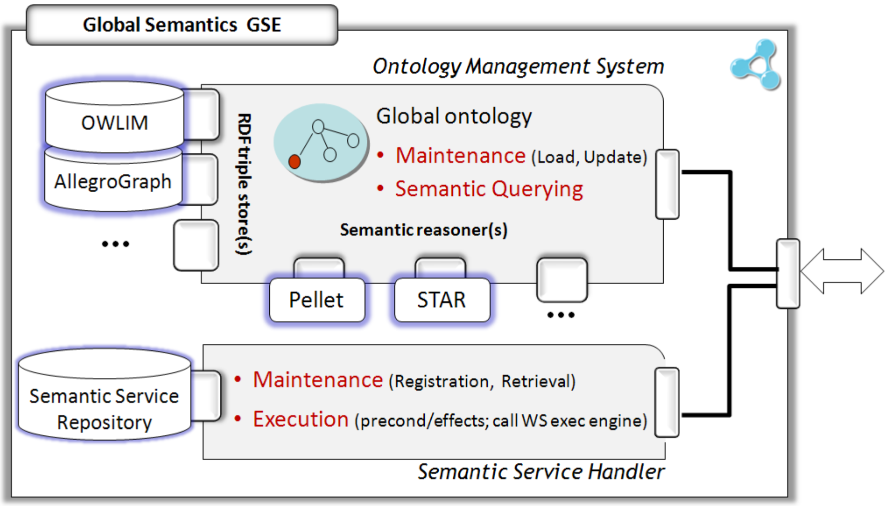


Fig. 4. Architecture of the global semantics environment (GSE).

Implementation. The implemented GSE has two architectural key features. First, its OMS has an open plug-in (API) architecture for using any RDF/S store and semantic reasoner as appropriate and is realized with the LarKC platform[10]. The OMS query decider routes semantic queries to OMS plug-ins available for the RDF triple stores SwiftOLIM (with RDF materialization of OWL2 under OWL-Horst semantics) and AllegroGraph, the semantic OWL-DL reasoner Pellet[11] with internal Jena RDF store, and the RDF relational reasoner STAR[14]. Second, semantic query answering and service handling by the GSE is upon request only, in particular, the GSE does not actively communicate semantic updates of the global ontology to other components; this avoids communication bottleneck and supports the paradigm of perception-based knowledge for BDI agents (cf. Section 4).

Semantic 3D Scene Query Processing. As a result of its open plug-in architecture, the types of semantic queries the OMS is capable of answering depends on the respective functionality of its plug-ins for triple stores and semantic reasoners. For example, the OMS can (a) efficiently answer object (and OWL-Horst concept) queries with its RDF store SwiftOWLIM using SPARQL, (b) more complex (OWL2-DL) concept queries with Pellet using SPARQL-DL, and (c) relational object queries with STAR. For example, a relational object query like ”How are scene objects doorAB, doorBC and roomC related ?” is processed by STAR by reduction to the corresponding NP-hard Steiner-Tree problem for the RDF graph of the materialized global ontology followed by the polynomial computation of an approximated solution in $O(nlogn)$ in terms of minimal RDF

---

[10] http://www.larkc.eu/resources/

[11] http://clarkparsia.com/pellet

object property-based path [14]. The pattern-based conversion of the result by our STAR-plugin of the OMS eventually yields a more human-readable answer (rather than just a list of RDF triples) like ”doorAB leads to roomB from where doorBC leads to roomC.” The semantic query decider of the OMS distributes semantic queries to the specific plug-in for processing based on the respective query type.

Global Semantic Service Execution Handling. Each semantic service registered at the global repository is executed either directly by the GSE or by the 3D graphics environment. In the first case, a service grounding updates the fact base without any 3D animation, while in the second case a grounding triggers both the 3D animation of an object and the change of its factual semantics in the global scene ontology such as the opening of a previously closed door.

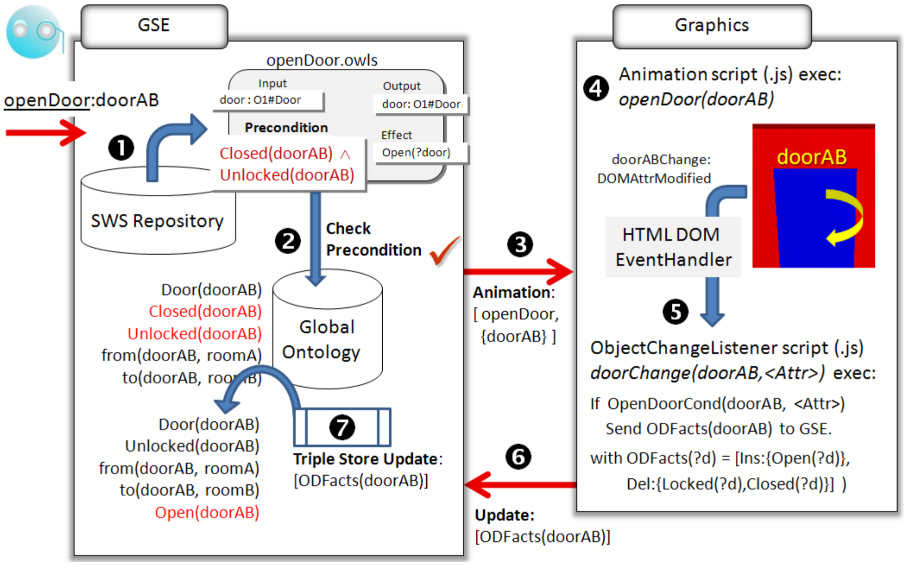


Fig. 5. Example of semantic 3D animation service execution handling by the GSE and 3D graphics.

Figure 5 illustrates the execution of semantic services with precondition and grounding in animation that effects the global ontology. In this example, the SemSH of the GSE receives the call of a semantic service ”openDoor” for opening the scene object ”doorAB” by some avatar (or its user). After retrieving the OWL-S description from its repository, it checks whether the service precondition (in SWRL) holds in the fact base of the OMS with a respective SPARQL ASK query. If successful, the SemSH triggers the execution of the service grounding, that is the 3D animation script for opening the door ”doorAB” in the scene by the graphics environment in the ISReal client. This animation may result in a change of the attribute values of the (HTML-DOM) XML3D scene graph object "doorAB" which is observed by an object script doorChange() through its registration at the HTML-DOM event handler for this object. In case the change has been encoded in the object script by the scene developer to correspond with a change of the animated object semantics in the global fact base, the script sends a semantic update query with respective insert-delete lists of object facts to the GSE for updating the global fact base by the OMS.

4 ISReal User Agent

In ISReal, the intelligent behavior of any avatar apart from direct user commands is determined by an intelligent agent it is uniquely associated with in the considered virtual world. The avatar represents the appearance of its user as her alter-ego but also its agent in the virtual world, and as such it is described as just another scene object in the XML3D scene graph of this world. The idea is that the user does not distinguish between her avatar and the intelligent agent that is driving its intelligent behavior; only in this sense, the terms "avatar" and "agent" can be used interchangeably. But how to design such an intelligent agent that is capable of understanding the semantics of the simulated scene it is involved in and to perform semantic-based action planning?

4.1 Architecture

For the development of ISReal agents, we adopted the reactive BDI (belief-desire-intention) architecture [22] that is known to be particularly appropriate for fast perception, deliberation and action in dynamically changing environments such as virtual 3D worlds. In very brief, the BDI agent is equipped with a plan library of domain-dependent and -independent plan patterns and a BDI planner to satisfy given goals by reactive action planning from second principles and execution. Based on the perception of its environment, that is the simulated 3D scene it is involved in, and given task to pursue, an ISReal agent selects an appropriate BDI plan pattern of operators, instatiates it with variable bindings in its local fact base into actions which execution in turn affect the perceived and locally updated state of the scene. Figure 6 outlines the architecture of an ISReal agent which consists of (a) an uniquely associated avatar that is running in the graphics environment (b) a semantic perception facility that interacts with an agent sensor running in the graphics environment and the GSE, (c) a BDI plan library and BDI planner, and (d) a local semantics environment (LSE) that maintains its local knowledge about the virtual 3D world it is involved in.

Local Semantics. The LSE of an ISReal agent differs from the GSE in several aspects. Though the local TBox is a copy of the global one copied from the OMS of the GSE during scene initialisation at the ISReal server, the local fact base includes only facts about scene objects the agent individually perceives via its sensor (cf. Section 4.1). The local service repository includes semantic services each of which encoding a plan operator from its BDI plan library; only services

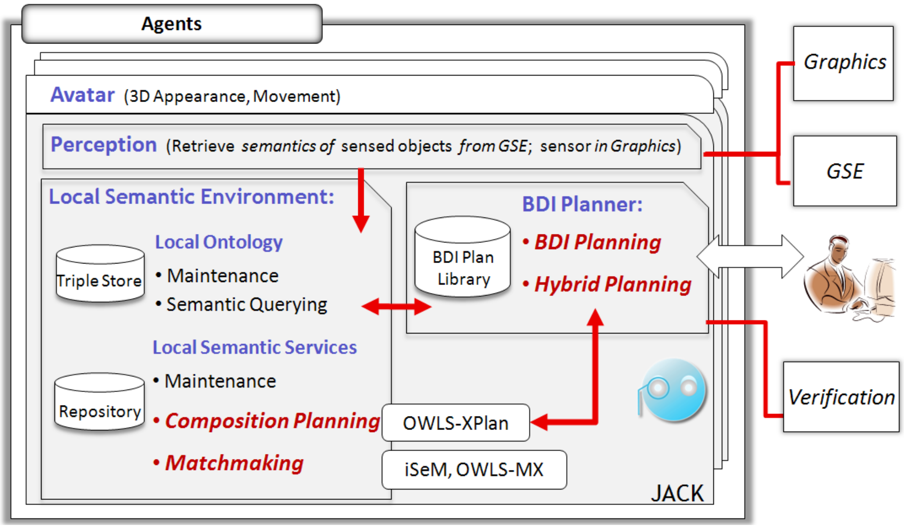


Fig. 6. ISReal user agent architecture.

with an effect on the global fact base of the scene are also registered with the global repository of the GSE. Further, the local SemSH handles the execution of agent services (not registered with the GSE) which groundings have no effect on the global fact base. This avoids that agents arbitrarily execute animations without checking the correlated change in the global scene semantics - such as walking through closed doors. Further, the LSE also offers plug-ins for local selection and composition planning of semantic services registered with the local repository.

4.2 Semantic Perception and Action Planning

Semantic Perception of 3D Scene Objects. An ISReal agent perceives its environment via its uniquely assigned agent sensor that is continously running in the 3D graphics environment. Each sensor is individually configurable regarding the frequency, resolution and range of object sensing. It sends the set of previously unknown or semantically updated XML3D scene graph objects (including the avatar object for self-perception) in the specified range together with their hashed metadata record (from the scene graph) to the individual agent. The semantics of perceived objects are then requested by the agent from the GSE which returns the set of (a) all object-related facts (corresp. with terminological object abstraction) from the global fact base and (b) all semantic object service descriptions. The subsequent update of the local fact base and registration of object services with the local repository completes the semantic perception. As a result, an ISReal agent only knows about those parts of the scene it perceives such that its local fact base may be inconsistent with the global one hosted by the OMS of the GSE. In particular, individual sensors or local update strategies (by default: immediately) of different agents may lead to different views on the global scene.

Semantic-Based Action Planning and Query Processing. In principle, an ISReal agent can answer the same type of semantic queries over its local ontology as the GSE over the global ontology of the simulated world. In addition, it can satisfy action goals like "Go to next room." or declarative goals like "Show me how to produce A with machine X?" by use of its BDI planner and semantic service composition planner over its local ontology and service repository. By default, an ISReal agent is equipped with domain-independent BDI plans for acting in virtual worlds such for basic 3D animations of its avatar, processing of different types of semantic queries in its local ontology, and execution handling of its actions (services) via the GSE or by itself. Other domain-dependent plans have to be added or customized for 3D scene simulation by the 3D scene agent developer.

In case there is no BDI plan pattern in the library that can be instantiated to satisfy a given action goal, the agent tries to solve this problem by action planning from first principles, that is the application of a semantic service composition planner to a given initial state in order to reach a given goal state. The initial state is a local fact base copy, the set of actions are and the goal state is either explicitly given or derived from the BDI plan context conditions at run time. Alternatively, the agent may search first for query objects missing in the local fact base by reactive BDI action planning before subsequent semantic service composition planning can be performed.

Implementation. The implemented agent environment is hosted by the ISReal server and consists of one intelligent ISReal agent and avatar by default, a model-driven BDI agent development tool, ISReal agent configuration tool for avatar assignment to and initialisation of an agent in a given scene, and the BDI agent execution platforms JACK and JADEX for server-sided running of the agent with client-sided 3D simulation of its avatar. The agent plug-ins for semantic service selection and composition are OWLS-MX, iSeM and OWLS-XPlan 2.0[12].

5 Use Case Example

The implemented ISReal platform 1.1 has been used to develop and simulate several virtual 3D worlds. In this section, we demonstrate this by means of a simple example for 3D simulation of production lines.

Small Virtual World "SmartFactory". The small virtual world "SmartFactory" consists of two rows of three rooms each (rooms A to C, D to F) where seven doors connect adjacent rooms, and one automatic ampoulle filling station X located in room F. The station is capable of filling different types of pills into a RFID-tagged cup that is placed on a transport wagon circulating between different pill production stations on demand. The filling state of the cup is read

---

[12] http://www.dfki.de/-klusch/i2s/html/software.html

via RFID sensors of designated control points while filling tasks are saved on its RFID chip. There is one default user avatar "Nancy" initially placed in room A. Figure 7 shows the layout and a screenshot of the user interface in the XML3D Chrome browser with the user avatar in front of the filling station.

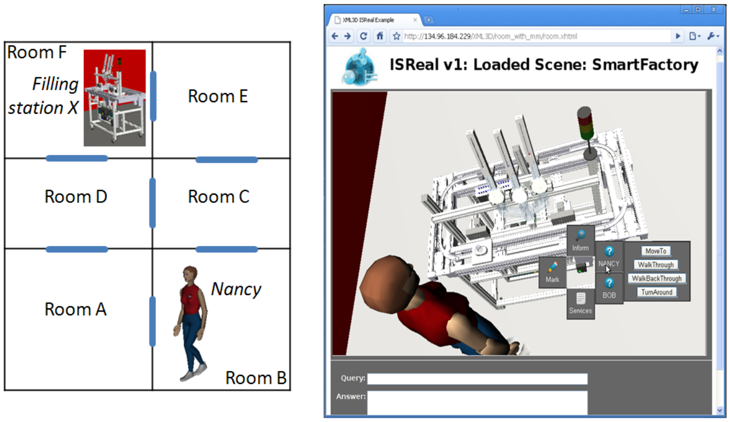


Fig. 7. Small virtual world SmartFactory: Layout and Web user interface snapshot.

Implementation. We graphically modelled this world including the avatar Nancy by using 3DSMAX and stored the designated HTML scene page with embedded XML3D description of the initial scene graph at the ISReal Web server. The global ontology for this world was developed by using Protege and then used for semi-automated semantic annotation of objects. The ontology defines 89 concepts, 38 properties with initially 57 facts about 38 semantically annotated scene objects (7 doors, 1 avatar, 1 station with 29 parts) while the global repository includes 29 object services in OWL-S. The ISReal agent for the single user avatar Nancy is modelled as a novice which knows nothing about the world objects: Its local fact base is empty and no object services are registered with the local repository yet.

Integrated Example of Intelligent Simulation in a Training Scenario. Once loaded into the XML3D browser client, the user can command her avatar to explore the scene and to demonstrate the functionality of the filling station on request. Consider the simple training scenario in which the user asks her avatar "show me how to produce a pill XYZ with the filling station X?". To satisfy the declarative goal by semantic object service composition planning, the agent first searches for the unknown query (goal state) objects pill XYZ, station X by reactive BDI action planning with in-room navigation, semantic perception of objects and respectively incremental update of the local fact base during its search. The subsequently generated service composition plan with OWLS-XPlan over the local ontology and 11 object services in the local repository is then reported to the user in form of text as an answer and then executed by corresponding 3D object service animations in the XML3D browser for demonstration.

6 Performance Evaluation

In this section, we discuss results of our preliminary performance evaluation of the ISReal platform 1.1 with focus on the global semantics and agent environment for the use case. The tests were performed with an average resourced notebook (Intel Quad Core Q9400, 2.66GHz, 8GB RAM) [13].

Global semantics environment. The results of our performance evaluation of the OMS plug-ins (SwiftOLIM, AllegroGraph, Pellet) for LUBM benchmark essentially are in compliance with those reported by others elsewhere [18, 16] and at the RDF Store Benchmarking site[14]. For our small use case, the triple store contained 1087 (747 explicit) triples. Figure 8 summarizes the thruput of the GSE (number of operations per simulation time). It shows, in particular, that within 1 second of simulation in the use case, the GSE can perform 15 updates and 75 queries over its store (precondition checks, fact retrieval; cf. Section 3). The STAR reasoner moderately scales up to 35k triples with 35s average query response time (AQRT).

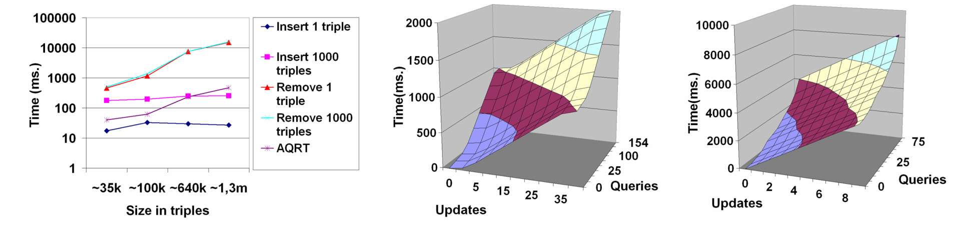


Fig. 8. Avg ontology update time for use case (1k triples) and LUBM(1, 5, 10) [Left]; AQRT in relation to updates for use case [Middle] and LUBM(35k) [Right].

Semantic perception by agent for use case. The agent needs avg. 80ms (320ms) to semantically perceive a scene object without (with) any annotated service. Semantic perception time is the period from received sensor perception event until completed update of the LSE with ontology and repository (cf. Section 4.2). Object fact retrieval from the GSE takes 6.5ms, 74ms for local fact base update and 240ms for registering object services in the local repository.

Semantic-based action planning by agent for use case. Preparation of the planning domain for offline semantic service composition planning using OWLS-

---

[13] Using the alternative immersive VR user interface of ISReal requires substantially more resources for reasonable 3D simulation performance (e.g. 44 Cores with 16GB RAM for about 15 frames per second).

[14] http://esw.w3.org/RdfStoreBenchmarking

XPlan takes 400ms and plan generation without execution (3D animations) takes 5s. The plan execution (service grounding in animation) appears fast with about 530ms: Precondition check per service by the GSE in 6.5ms, 3D animation by the graphics environment in 300ms and update of the global fact base in 27ms.

Discussion. Semantic-based 3D simulation with ISReal appears reasonable for small virtual worlds with medium-sized global scene ontology, few hundreds of annotated 3D scene objects and up to a few dozens of agents. For example, the simulation time of 30 agents concurrently trying to open 30 doors would take only 1.5 seconds: Checking of 30 preconditions by the GSE (195ms), concurrent animation of door opening (500ms) and 30 updates of global fact base (810ms). The GSE slows down the overall 3D simulation for one agent with 33.5ms and for 30 agents with 1 second. The reactive BDI action planning appeared extremely fast (avg 25ms per plan operation check) but the main slow down of semantic-based planning within simulation is caused by semantic service composition with 5 seconds.

7 Related Work

It appears common sense that the use of semantics can greatly improve the management and retrieval of 3D content [11, 19, 21, 10] as it has been impressively demonstrated for various practical applications in different domains such as arts, bioinformatics, gaming, cultural heritage and virtual museums, partly in relevant projects like Aim@Shape, FocusK3D and 3DVisa[15]. To the best of our knowledge, the open 3DI platform ISReal significantly differs from this body of work in general: Semantic annotation of 3D scenes in standard RDFa and OWL2 enables users and avatars alike to better understand the semantics of simulated 3D objects and their relations in the virtual world; the representation of 3D scene graphs in XML3D allow for all-in-one and highly realistic 3D scene rendering by any XML3D browser with our real-time raytracer RTFact; any ISReal avatar is potentially capable of behaving more intelligent than other types of avatars in virtual worlds available today thanks to the capabilities of its associated intelligent agent for semantic reasoning, semantic-based action planning and service composition.

In particular, many approaches to integrate semantic Web with virtual 3D worlds [20, 12] put a strong emphasis on exploiting semantic 3D content annotation in RDF, RDFS or proprietary formats for semantic object search and querying in specific application context such as the virtual 3D furniture shop in [8, 15] or the virtual 3D museum tour guide in [4] - but without semantic-based action planning by user avatars like in ISReal. On the other hand, related work on virtual agents such as in [1, 3] and STEVE (SOAR Training Expert for Virtual Environments)[23] focus rather on multi-modal user-agent interaction in immersive VR environments and use of AI planning by agents with pre-coded planning domain knowledge and plan patterns - but without any semantic-based 3D scene querying or service composition planning from first principles like in ISReal.

---

[15] http://3dvisa.cch.kcl.ac.uk/project86.html

8 Conclusions

We presented the first open 3DI research platform for semantic-based 3D simulations in virtual worlds that uses semantic Web, semantic services, intelligent agents and 3D graphics. ISReal user avatars are coupled with intelligent agents that understand the semantics of their annotated 3D environment and perform semantic-based action planning to satisfy goals (queries) of their users. Such intelligent 3D simulations with the implemented platform are reasonably fast for small virtual worlds with medium-sized global ontology, small number of annotated 3D scene objects and up to a few dozens of agents.
However, scalability of semantic 3D scene query processing and semantic service composition planning remains an issue for intelligent simulation of time-critical applications of large virtual worlds with potentially thousands of semantically annotated scene objects and hundreds of agents. Our ongoing work for ISReal 2.0 is on multi-agent planning scenarios, multi-user applications, and scalable semantic query processing respecting relevant work [26, 28] and research results from LarKC and SEALS[16]. The implemented ISReal platform 1.1 together with the SmartFactory use case will be released under GPL license.

References

1. Abaci, T.; Ciger, J.; Thalmann, D. (2005): Planning with Smart Objects. WSCG SHORT papers proceedings, UNION Agency-Science Press.
2. Alpcan, T.; Bauckhage, C.; Kotsovinos, E. (2007): Towards 3D Internet: Why, What, and How? IEEE Intl. Conference on Cyberworlds. LNCS 2174, Springer.
3. Anastassakis, G.; Ritchings, T.; Panayiotopoulous, T. (2001): Multi-agent Systems as Intelligent Virtual Environments.
4. Chou, S.-C.; Hsieh, W.-T.; Gandon, F.L.; Sadeh, N.M. (2005): Semantic Web Technologies for Context-Aware Museum Tour Guide Applications. Proc. 19th Intl. IEEE Conf. on Advanced Information Networking and Applications.
5. Daras, P.; Alvarez, F. (2009): A Future Perspective on the 3D Media Internet. Towards the Future Internet - A European Research Perspective, IOS Press.
6. de Silva, L.; Sardina, S.; Padgham, L. (2009): First Principles Planning in BDI Systems. Proc. 8th Intl. Conf. on Autonomous Agents and Multiagent Systems.
7. Davies, N., Mehdi, Q. (2006): BDI for Intelligent Agents in Computer Games. Proc. 8th Intl. Conf. on Computer Games.
8. De Troyer, O.; Kleinermann, F.; Mansouri, H.; Pellens, B.; Bille, W.; Fomenko, V. (2006): Developing semantic VR-shops for e-Commerce. G.D. Magoulas, G. Lepouras, C. Vassilakis (eds.): Special Issue of Virtual Reality: Virtual Reality in the e-Society, Springer.
9. Georgiev, I.; Rubinstein, D.; Hoffmann, H.; Slusallek, P. (2008): Real Time Ray Tracing on Many-Core-Hardware. Proc. 5th INTUITION Conf. on Virtual Reality.
10. Geroimenko, V.; Chen, C. (2005): Visualizing the Semantic Web: XML-based Internet and Information Visualization. Springer.
11. Ibanez-Martinez, J.; Mata D. (2006): Virtual Environments and Semantics. European Journal for the Informatics Professional, 7(2).

---

[16] www.seals-project.eu

12. Ibanez-Martinez, J.; Mata, D. (2006): A Basic Semantic Common Level for Virtual Environments. Virtual Reality, 5(3).
13. Kallmann M. (2001): Object Interaction in Real-Time Virtual Environments. PhD thesis, École Polytechnique Fédérale de Lausanne.
14. Kasneci, G.; Ramanath, M.; Sozio, M.; Suchanek, F.M.; Weikum, G. (2009): STAR: Steiner Tree Approximation in Relationship-Graphs. Proc. 25th IEEE Intl. Conf. on Data Engineering.
15. Kleinermann, F.; et al. (2008): Adding Semantic Annotations, Navigation Paths and Tour Guides to Existing Virtual Environments. LNCS 4820, Springer.
16. Kiryakov, A (2008). Measurable Targets for Scalable Reasoning. EU FP7 Project LarKC, Deliverable D5.5.1.,www.larkc.eu/deliverables/
17. Kleinermann, F. (2007): Adding Semantic Annotations, Navigation Paths and Tour Guides for Existing Virtual Environments. Proc. 13th Intl. Conf. on Virtual Systems and Multimedia, Springer.
18. Luther, M.; Liebig, T.; Bhm, S.; Noppens, O. (2009): Who the Heck Is the Father of Bob? Proc. 6th European Semantic Web Conference, Springer.
19. Ontology Schema (RDFS) for a subset of the X3D language. http://cs.swan.ac.uk/-csowen/SVGtoX3D/examples/X3D OntologyRDFS.htm
20. Pittarello, F.; De Faveri, A. (2006): Semantic Description of 3D Environments: A Proposal Based on Web Standarts. Proc. Intl. Web-3D Conf., ACM Press.
21. Polowinski, J. (2007): SemVisHistory. Visualisierung von Semantic-Web-Geschichtsdaten in 3D. Available: netzspannung.org/database/401953/de
22. Rao, A.S.; Georgeff, M.P. (1995): BDI Agents: From Theory to Practice. Proc. Intl. Conf. on Multi-Agent Systems (ICMAS), AAAI Press.
23. Rickel, J.; Johnson, W.L. (2001): Task-oriented collaboration with embodied agents in virtual worlds. Embodied conversational agents, MIT Press.
24. Rubinstein, D.; Georgiev, I.; Schug, B.; Slussalek, P. (2009): RTSG: Ray Tracing for X3D via a Flexible Rendering Framework. Proc. 14th Intl. Conf. on Web3D Technology, ACM Press.
25. Smart, J.; Cascio, J.; Paffendorf, J. (2007): Metaverse Roadmap - Pathways to the 3D Web. www.metaverseroadmap.org/MetaverseRoadmapOverview.pdf
26. Soma, R.; (2008): Parallel Inferencing for OWL Knowledge Bases. Proc. 37th Intl. Conf. Parallel Processing, IEEE.
27. Sons, K.; Klein, F.; Rubinstein, D.; Byelozyorov, S.; Slusallek, P. (2010): XML3D Interactive 3D Graphics for the Web. Proceedings of Web3D 2010 Conference.
28. Urbani, J.; Kotoulas, S.; Oren, E.; van Harmelen, F. (2009): Scalable Distributed Reasoning using MapReduce. Proc. 8th Intl. Semantic Web Conference, Springer.
29. Walczak, A.; Braubach, L.; Pokahr, A.; Lamersdorf, W. (2006): Augmenting BDI Agents with Deliberative Planning Techniques. Proc. 5th Intl. Workshop on Programming Multiagent Systems.
30. Warwas, S.; Hahn, C. (2009): The DSML4MAS Development Environment. Proc. 8th Int. Conf. on Autonomous Agents and Multiagent Systems.
31. Zahariadis, T.; Daras, P.; Laso-Ballesteros, I. (2008): Towards Future 3D Media Internet. NEM Summit 2008, St. Malo. http://www.ist-sea.eu/Dissemination/SEA_FIA.pdf