

A research on an intelligent multipurpose fuzzy semantic enhanced 3D virtual reality simulator for complex maritime missions

Ahmad C. Bukhari · Yong-Gi Kim

Published online: 8 July 2012
© Springer Science+Business Media, LLC 2012

Abstract In this article, the idea of multipurpose fuzzy semantic enhanced 3D virtual reality simulator for the evaluation of maritime robot algorithms, and for the analysis of maritime missions is presented. The simulator uses the digital mockup technology in blending with semantic domain knowledge of the system, to analyze the tasks remotely. The 3D virtual reality (VR) is proposed to help the operators provide the detailed information and guidance during real-time tele-operations, and the incorporation of fuzzy semantic knowledge makes the virtual environment intelligent and automatic. The integration of fuzzy scene independent ontology with the virtual environment (VE) engenders a knowledge driven inter operable virtual environment which eases the user in natural language querying, personalization, interpretations and manipulation. The distinctive semantic VR scene builder utility of the proposed system draws the VR environment automatically while getting the high level specification of the system for the client. The proposed simulator can be effectively used for real-time robots trainings and for the evaluation of AI based algorithms designed for intelligent vessels and AUV's without knowing the complex underlying VR scene building technologies. Furthermore, it provides the benefits to optimize the pre-process physical environment operations to mimic the real world into a virtual environment. The remote operations and feasibility analysis performed on virtual simulator can efficiently save the cost,

time and claims to provide the operators a preprocessor information and guide.

Keywords Fuzzy semantic marine simulation · Semantic virtual reality · Automatic marine simulation · 3D VR simulation for autonomous vessel and AUV · Scene independent fuzzy ontology

1 Introduction and related research work

The marine robots such as AUV's (Autonomous Underwater Vehicles), ROV's (Remotely Operated Vehicles) and autonomous vessels gained popularity in recent years due to their special attributes of performing labor intensive repetitive and precise information gathering tasks, in hazardous environment without human supervision. The AUV cruises under sea according to maneuvering scenarios while fetching data and avoiding obstacles. It is very difficult to gather complete information about unpredictable environment and tackle the issues which arise in non-tethered system [1, 12]. The increasing complexities of marine operations demand to design and develop more efficient control algorithms and architectures. The development and performance analysis of marine robot algorithms comprises of several complex stages and to carry these out, we have to go through several real-time experiments. The open sea real-time experiments need a highly skilled team with a complex logistic. Usually, it is impossible to perform the repeated and infrequent open sea tasks due to unpredicted and time varying environment [22, 25]. The use of simulation plays an important role in the algorithm testing in a short time span and in a cost effective way.

Virtual reality is gaining popularity as an effective engineering design tool because of its intuitive interaction with

A.C. Bukhari · Y.-G. Kim (✉)
Artificial intelligence Lab, Department of Computer Science
and Engineering Research Institute (ERI), Gyeongsang National
University Jinju, Kyungnam 660-701, Korea
e-mail: ygkim@gnu.ac.kr

A.C. Bukhari
e-mail: ahmadchan@gmail.com

computer-generated models. The immersive aspects of virtual reality provide more sophisticated tools to interact with dynamic components of virtual environment. Currently, the researchers are performing their best in the development of VR based test bed to provide the cost effective alternate of physical manipulation [7]. In the last two decades, the researchers have presented several solutions based on a simulation which can decrease the marine robots experimental time and cost [27]. At the early stage of marine robots design and algorithm validity testing, offline simulators can be considered as an effective solution. MATLAB/SIMULINK is a very popular offline simulator in the researchers' community now a day. Although, the offline simulators can evaluate the internal working of marine robots efficiently; yet there are the couple of technical limitations in offline simulators which decrease the popularity graph of offline simulators. Usually, the maneuverability of marine robots is directly linked to external sensors and the offline simulators cannot interact with external sensors due to technical limitations, and consequently, fail to produce the external world [12, 16].

Thus, the researchers feel difficulty while testing the complex maritime algorithms using offline simulators. On the other hand, the online simulators provide the concurrent solution to draw the external world based on external sensor inputs. The third type of simulators is known as HIL (Hardware in Loop) simulators. The HIL simulators amenities the client to examine the designed algorithms on actual hardware, the HIL simulator gets the input through on board sensors and route the robot manipulation towards the simulation world. This hybrid mechanism of HIL simulator combines both the offline and online simulators' properties and therefore, provide the detailed analysis of the marine robot's algorithms. To reduce the design and implementation timing of AUV development, research on multi vehicle real-time graphical simulator is presented in [33]. The Neptune is the both hybrid and hardware in the loop simulator which closes the gap between offline simulation and real-time testing by providing a graphical interface based on OpenGL.

The Neptune simulator takes the three files as input to design virtual testing environment. The first file is the VRML file which holds the information about robots and environment. The second file contains the robots and the thruster hydrodynamic coefficient information and the third and last file carries the information about the previous two files along with information about sensors. In 1990, the naval postgraduate school of united state developed an experimental AUV to analyze the possibility of autonomous robots in wars. The NPS AUV-SIM-2 [28] is based on knowledge base and intelligent algorithms to expedite the development and deployment process of the AUV sub systems. The efficiency of NPS AUV-SIM-2 is considered to be high as it saved a lot of

time during the NPS's AUV research project [49]. In [12], the idea of integrated simulation for rapid development of the AUV is presented. An integrated simulation communicates with the hardware components of the robots directly and supports the working of equipment with graphics simulation. This simulator was based on high resolution 3D graphics, and can present vehicle dynamics, control system behavior, mission execution and sonar processing graphically, which helps the researcher to better, evaluate the performance of the components.

At present, ontology has proved itself as an effective technology for relevant information extraction and knowledge representation. We used the ontology to store and exploit the domain knowledge for the development of virtual environment. This unique power of the ontology can be exploited to design an accurate knowledge representation system. During the past few years, researchers are using ontology widely in different domains for knowledge representation and information extraction. It is another fact that the most of the information resides in offline and online resources, is in imprecise format. The ontology only deals with crisp data and cannot find desired results from vague sources of data [35]. To address this issue, the researchers incorporated the fuzzy logic theory into classic ontology. The fuzzy theory is widely known among the researcher community due to its ability to deal with blur information [38]. The combination of both technologies has gained popularity and researchers have proposed several solutions based on fuzzy ontology. In [42] Sun Yi et al. proposed a fuzzy ontology based solution to represent the Chinese medicine. This paper highlighted the process of fuzzy ontology construction and its application to represent the Chinese medicine which helps to cope with liver diseases.

Jun Zhai et al. accommodated the fuzzy ontology based framework in their research [47] to deal with vague linguistic values of fuzzy concept; in supply chain management. The information sharing and retrieval is the core activity in supply chain management. The fuzzy ontology based framework makes the information acquisition and sharing handy in supply chain management. To identify the existing boundaries of knowledge in patent search decision support system, researchers suggested a solution in [34] by using the fuzzy ontology. Acceptation of any patent is based on its uniqueness; fuzzy ontology based software can help to find the similar domain patterns against any requested patent for editors. In [21] Huiying et al. used the ontology to develop enterprise information retrieval model. As the traditional information retrieval systems cannot understand the users' potential query requirements because of its keyword base structure, hence, the usage of ontology effectively works in such scenario. Qual Thanh et al. took one step forward and investigated a solution to automate the fuzzy ontology generation in semantic web [37].

The FOGA (Fuzzy Ontology Generation Framework) brought easiness and automation in manual construction of fuzzy domain ontology. In literature, we found multidisciplinary work of Jun Zhai and his team by using fuzzy ontology framework. They applied fuzzy ontology to extract the relevant information from e-commerce domain [46]. They also worked on knowledge modeling for fuzzy system based on ontology approach and presented their work in [45]. In this paper, we presented a unique idea for marine simulation based on 3D fuzzy semantic virtual reality due to several compelling reasons. All the marine simulators which we found in the literature are designed for expert users. Typically, the artificial intelligence algorithm developers and marine domain specialists do not have enough expertise to design the simulation environment through low-level programming to test their algorithms. Therefore, to test an algorithm, an AI expert has to hire graphics programmers or as an alternate has to subscribe the high price software. Since, there is a wider technical gap between algorithm designing and its performance judgment; thus, the researchers have to wait several months to exact judge the performance of their algorithms. We proposed fuzzy ontology supported mechanism in our research which narrows the barrier of manual simulation scene developments.

Its automatic NLQ based VR scene building utility takes the high level components of the experiment from the client, and intelligently plots the desired simulation environment. The hidden power of fuzzy scene independent ontology can extract the features from natural language queries and helps the user to inquire from the scene in natural language. The three dimensional environment which is generated by our proposed simulator also has a prominent feature comparing with other simulators. All the components in our simulation are dynamic and can interact with one another through XML. This feature helps the researcher to get the experimental results closer to the real-time experiments. Further, the dynamic mission planner module of our system can easily create an experimental scenario without knowing the complex background logic. We evaluated the performance of our proposed architecture by developing a Java based prototype and performed several experiments. To increase the reader's interest, we designed a real-time scenario with the help of marine engineer and applied the proposed solution step by step and displayed the results in the experiment section (Sect. 6). The organization of the rest of the article is; in Sect. 2, we presented our proposed research architecture. Sections 3 and 4 are explaining the internal working of the proposed solution in detail. Section 5 highlights the features of semantic virtual reality graphics. In the last section, we added the ended remarks and shared the potential points for future work.

2 The proposed research design

Maritime robots are playing significant roles in various maritime areas such as oceanography, maritime security, undersea exploration, pipelines maintenance and inspection. The increasing complexity in maritime operations is demanding for more sophisticated algorithms which can handle the complex tasks autonomously or semi autonomously with high precision [39]. The testing and gauging of intelligent algorithms and designing the real time scenarios to observe the efficiency of the algorithm is time consuming and costly [7, 33]. One possible solution to cope with this situation is to test the whole scenario through virtual testing with simulation. In recent years, the researchers developed several solutions based on virtual simulation; Sect. 1 is explaining some of these. However, to design a virtual testing environment according to domain expert's demand is also a hard task; which requires deep familiarity with virtual reality scene development technologies. The maritime scientists and the algorithm designers are mostly non-VR personals, so it is considered to be very hard for them to design a virtual test bed to evaluate their designed algorithms [24].

To expedite the process of algorithm testing and marine mission evaluation, the researchers are looking for such kind of solution which can automate the process of virtual reality scene building and mission planning automatically while taking the high level input from the operator. The basic theme of our proposed fuzzy semantic based 3D virtual reality simulator is to make the virtual experimental process automatic and more result oriented. The core of our proposed solution is based on fuzzy domain ontology which creates an intelligent knowledge base for the system. The simulator is proposed to be further equipped with the dynamic path planning and obstacle avoidance techniques resulting in a tool for pre analysis, and simulations for maritime robots' operations. This simulator can be used as a design tool for evaluating marine operations using virtual prototypes of realistic models. The proposed simulator can be designed for five major uses: visualization, basic concept development of equipment, design verification of marine robots, development and investigation of the optimum operation sequence of trajectory recording of the underwater path planned performed. Furthermore, complex real time operations can be carried out in the designed in the near future. Figure 1 shows the architecture of the simulator. We divided the architecture (Fig. 1) of the system into four functional layers for simplicity. In the next sections, we will elaborate the internal working of each layer one by one.

2.1 Client interaction layer (CIL)

The client interaction layer consists of three interactive components which are: VR scene builder, mission planner and

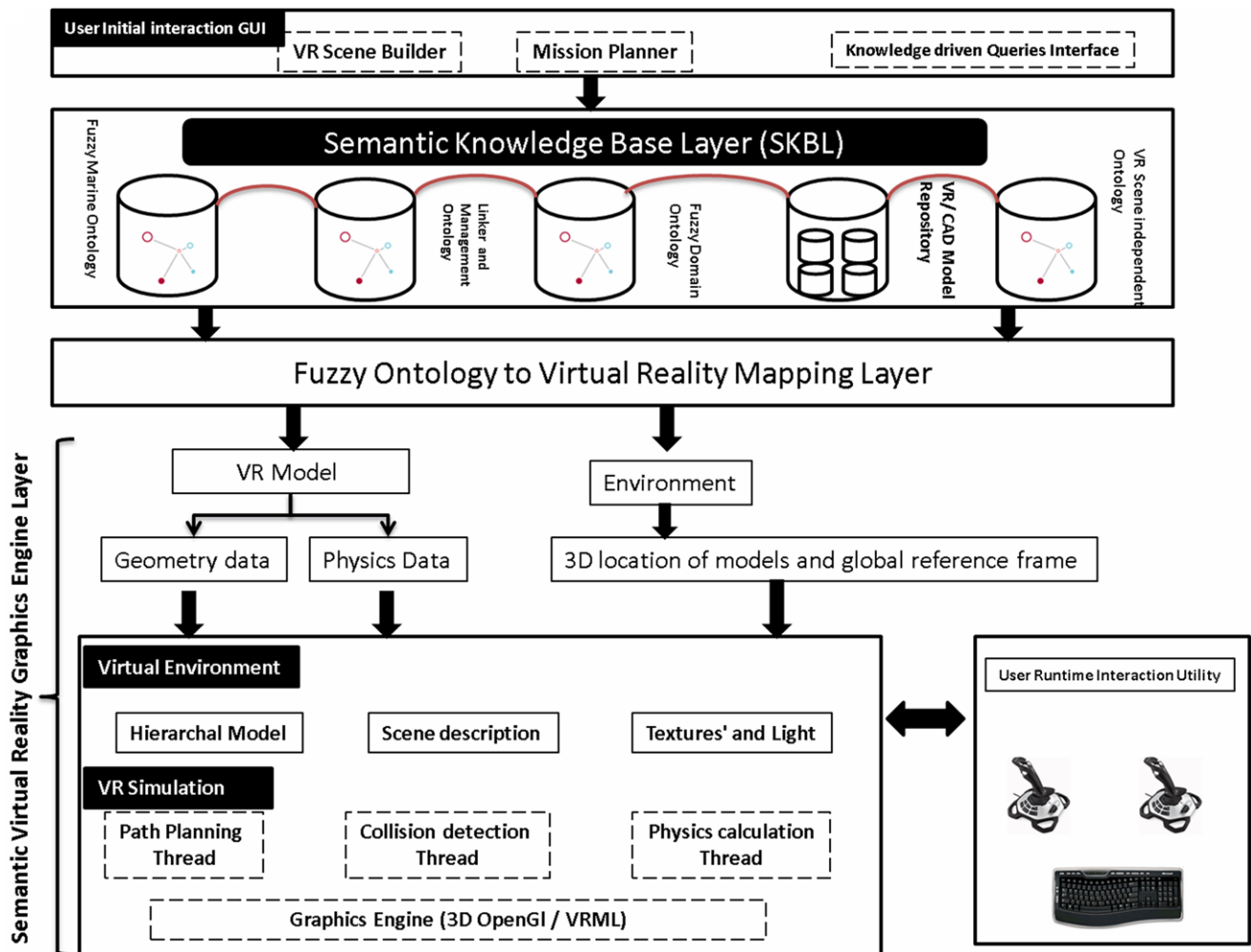


Fig. 1 The anatomy of fuzzy semantic assisted 3D virtual reality simulator

knowledge driven query interface. These three components are specifically designed to facilitate all types of user; from non-technical to expert users. The strength of CLI lies in its powerful natural language processing feature. By clicking on VR scene builder utility, we can find an interactive user input dialog which leads to the generation of 3D VR environment. The dialog first asks for the user about the mission type, there are two options which user can select in drop down list: undersea and sea surface. The selection of mission type loads the further form fields like marine robot type (which will be used in the experiment). As the simulator is designed for multipurpose operations thus it offers AUV, ROV, submarine, vessels, hovercraft and autonomous boats as an option to select. The proposed simulator provides leverage to the user to adjust the width and height of the VR scene according to the requirements (width and height can be adjusted in pixel or in centimeters (cm)). The scene type and light option can convert the virtual reality scene into two or three dimensional spaces. Figure 2 sequence diagram of natural language query processing.

The user can also set the background color or background texture according to experiment. The interactive VR scene builder also provides an option to edit the scene manually. This option can be helpful for expert users to modify the virtual environment manually to make the virtual test bed more requirement specific. Figure 4 is screening the VR scene building dialog. The second option of CIL is the virtual mission planner. The sophisticated mechanism of mission planner utility assists the client to define the mission in natural language. The mission planner asks the user to enter the mission requirements through multi dialog sessions. The user can set the start and end positions and can add the environment parameters like sea depth, light condition, rocky environment, etc. The mission planner also provides the option for manual editing of the scene for experimental algorithm integration. To get the output from natural language query (NLQ), we first translated the NLQ to structured SPARQL by using the techniques presented in [10, 14, 38]. For better understanding of the readers, we take a sample scenario here and perform the natural language processing.

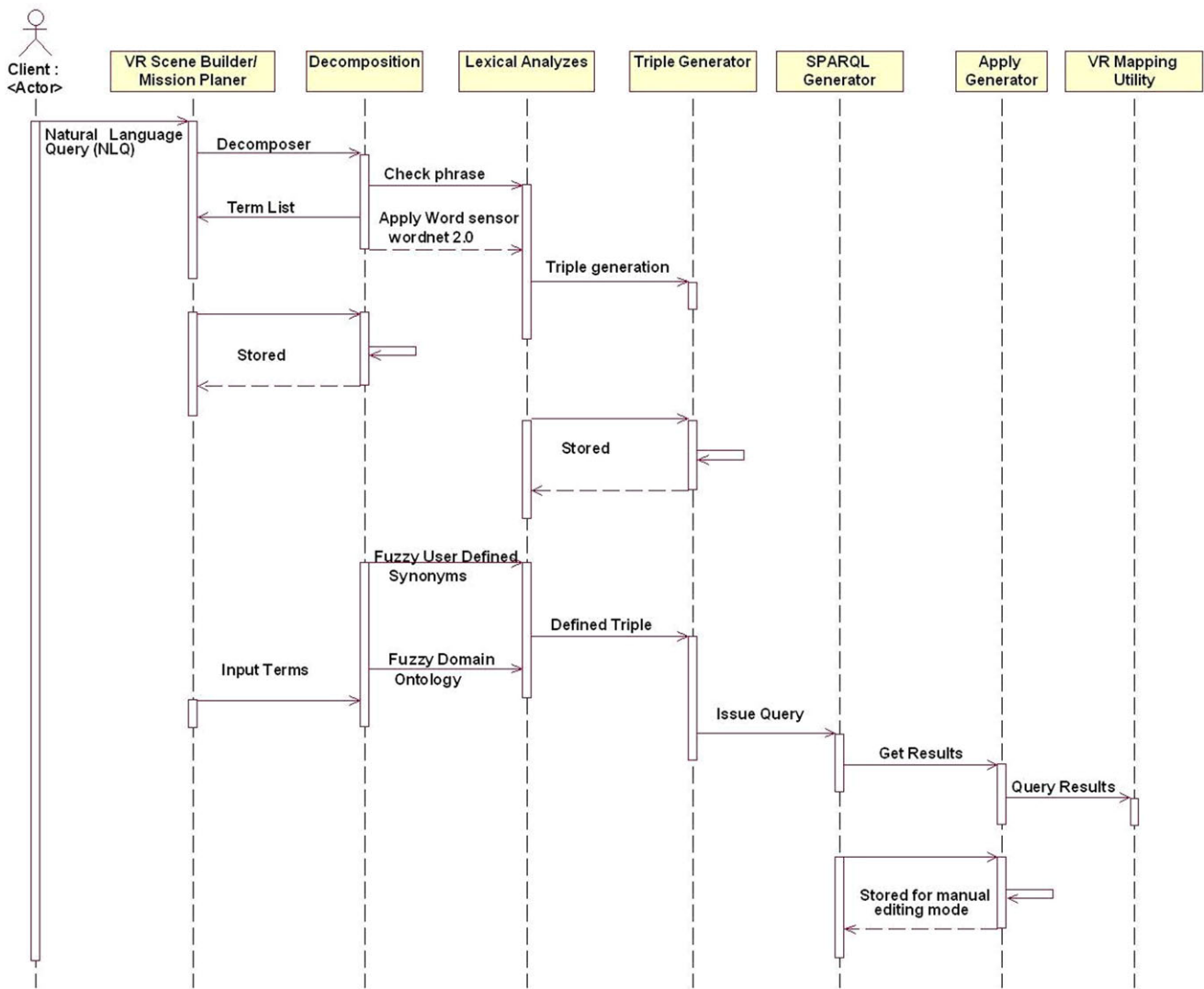


Fig. 2 Sequence diagram of natural language query processing

“Draw an underwater 3D environment with one AUV which has started a position (0.0, 0.0, 0.0), translation (0.0, 0.5, 0.0) and end position (12.0, 12.0, 0.0) and obstacle_1 with position (5.0, 5.0, 0.0), obstacle_2 with position (11.0, 5.0, -5.0), obstacle_3 with position (9.0, 8.75, -2.0). Translate AUV from start to end position and avoid the obstacles.”

The UML diagram in Fig. 2 expresses the natural language processing steps in sequence. The natural language processing field is full of challenges; however, the challenges in natural language to query translation can be categorized into two steps. 1—Natural language parsing and 2—SPARQL query translation. The arbitrary structure of the natural language makes the parsing process complicated and literature shows that even with a very refined language parser can parse up to 90 to 95 % data precisely. We utilized

the Stanford Parser¹ to handle the first issue. As described in Fig. 2, when a client entered the natural language query, the natural language processing function of the backend algorithm takes the input and decomposes the statements into small chunks with extraction of terms list. The Stanford Parser divides the chunks into categories and stored them for further use. In the next steps, the algorithm forwards the output to the lexical analyzing unit which analyzes the input and applies the refinement steps such as: stop word reduction, synonym matching, semantic relationship identification among terms, etc.

We used WorldNet 2.0,² fuzzy marine ontology and fuzzy user defined synonym repository in lexical analyzing step. As shown in Fig. 3, the algorithm searches for

¹<http://nlp.stanford.edu/software/lex-parser.shtml>.

²<http://wordnetweb.princeton.edu/perl/webwn>.

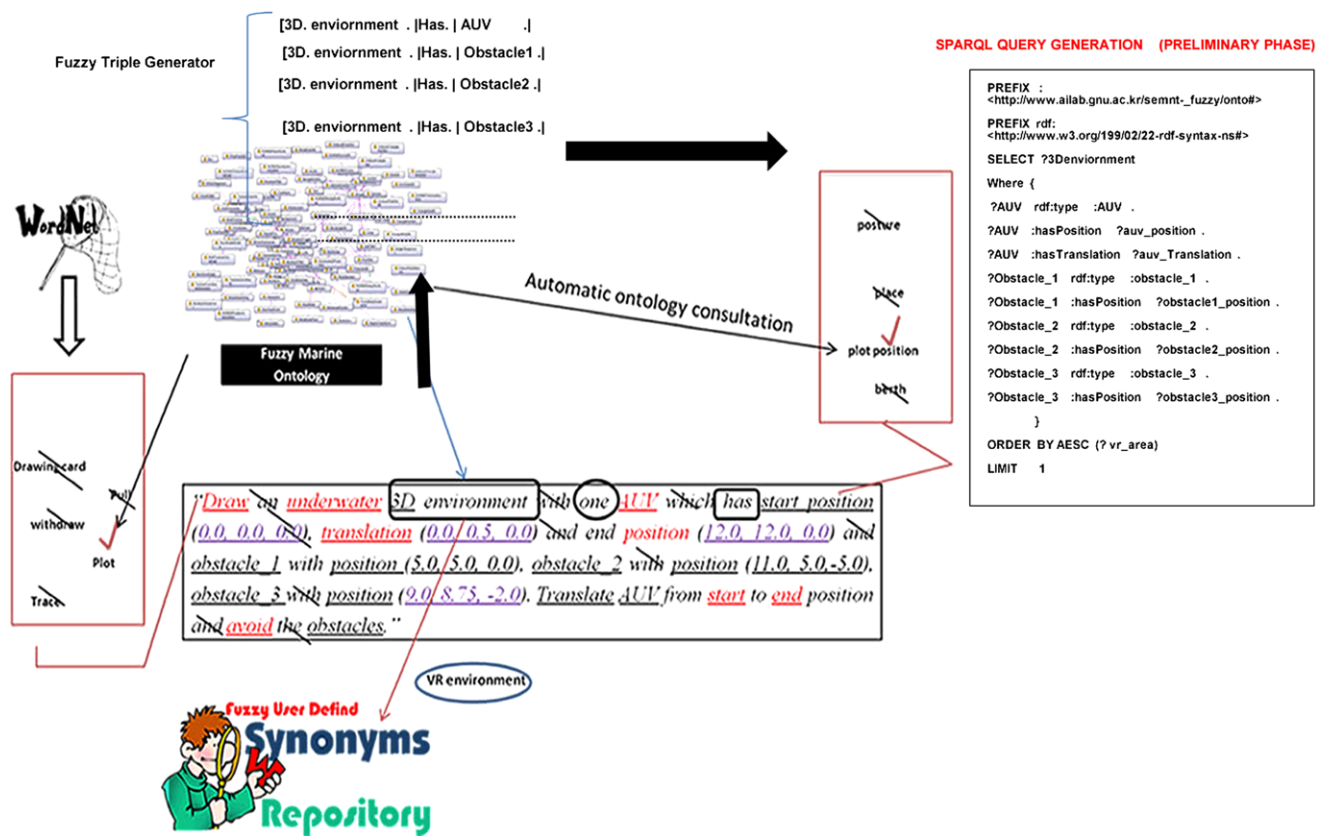


Fig. 3 A graphical view of arbitrary natural language query to SPARQL translation

appropriate terms after stop word reduction process by using WorldNet 2.0. During the term building and refining phase the algorithm continually consults the fuzzy marine ontology to extract the right terms. By taking the above scenario, our tool found synonyms of each word, in consultation with WorldNet 2.0. However, the lexical analyzing algorithm consulted the fuzzy marine ontology and fuzzy user defined synonyms to select the right potential terms to generate the language triples. In the next phase, the SPARQL generator takes the output of the last step as input and translated the language triples to SPARQL query [19]. The SPARQL query is the standard query language which is developed to extract the information from ontology particularly. In the last step, our proposed simulator passed the query to the fuzzy domain ontology and extract and forward the results to fuzzy mapping layer. It is also interesting to note, during the experiment phase several times, we had to refine the algorithms to get the desired results. The results section (Sect. 6), briefly explains the precision and recall statistics; which were generated during the experimental phase.

3 Semantic knowledge base layer (SKBL)

The second layer of proposed simulator is a semantic knowledge base layer (SKBL). The SKBL is the backbone of the

whole system; it helps to create the virtual test bed which mimics the real world. The SKBL consists of five components which are: fuzzy marine ontology, linker and management ontology, VR scene independent ontology, domain ontology and CAD/CAM model repository (combined component of layer 3 and 4). An ontology is an explicit formal specification of a shared conceptualization of any domain, which is machine readable and human understandable format [20]. Primarily, ontology is developed to share a common understanding of the domain knowledge among people and software, in order to reuse the classes of a domain instead of remodelling them [30, 40]. The domain ontology describes the knowledge of the domain as concepts, their properties and relationships between these properties and concepts. This section elaborates the modelling process of fuzzy ontology of VR based marine system. The literature review reveals that a few researchers worked on VR based maritime systems and those who worked; their work was limited to academic purposes which could not solve the problems of real world domains [2, 11]. The strength of our novel simulator is the fuzzy enriched domain ontology which not only describes the domain knowledge of marine system but also assists the operator to acquire the needed information from ontology using natural language queries. Ontology can be expressed in equation format as:

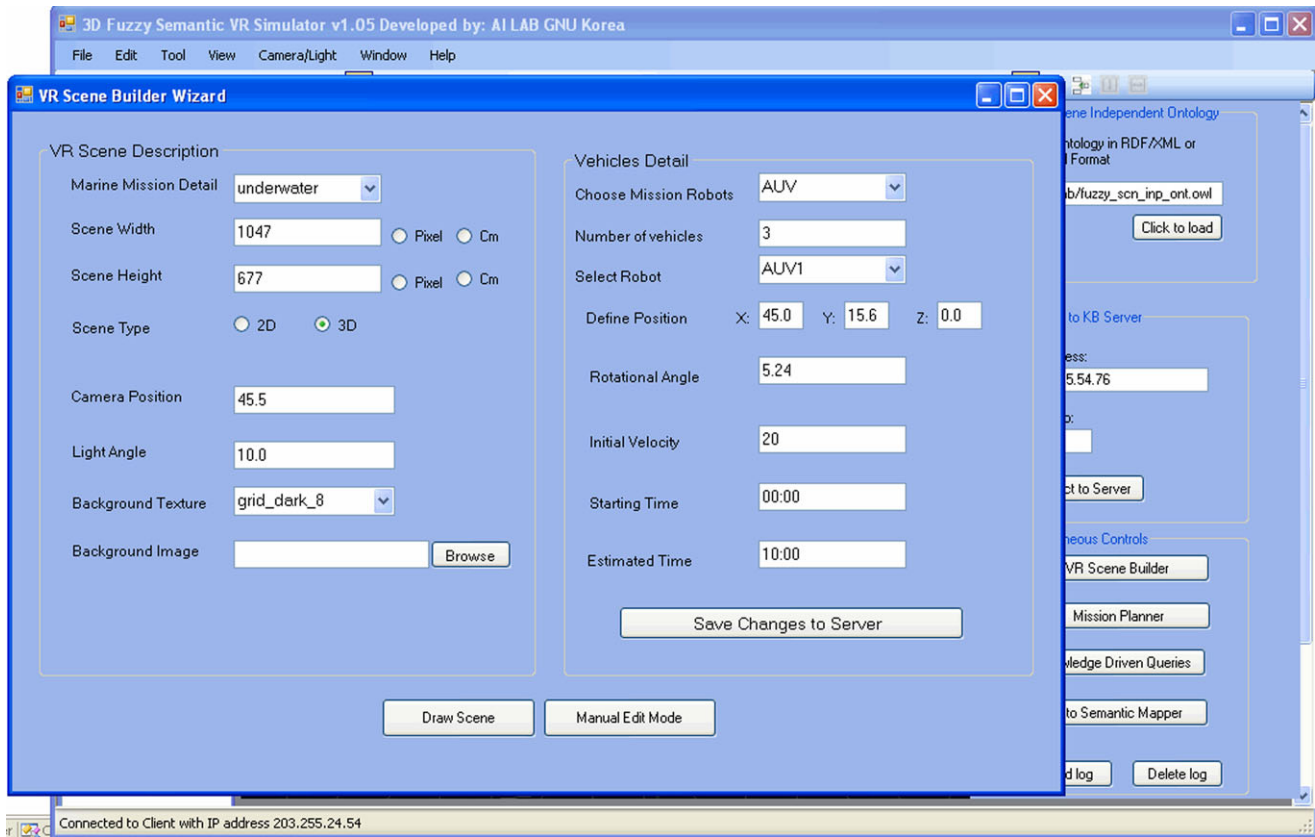


Fig. 4 The NLQ based interactive VR scene building utility of prototype

Ontology = (Concepts C , Properties P , Relationships R , Values V , Constraints on property value V_C) (1)

In the above expression, the notations C , P , R , V , V_C represent concepts, properties, and relationship among concepts, concept values and constraints on properties respectively. The W3C has developed a specific language to write the ontology called OWL (Web Ontology Language), OWL is considered to be a standard language for ontology modelling [4]. To increase the expressivity level in OWL, several new constructs introduced in OWL-2 along with their unique modelling by W3C engineers. Presently, the ontology has become the hot spot in the intelligent information engineering domain, and can be classified into meta ontology, domain ontology and application level ontology. We usually, exploit the domain ontology to represent the knowledge of a particular domain [2, 3]. On the other hand, there is not exactly one way to define an ontology, researchers introduced diverse ways to define the concepts of the domain in ontology according to their needs and demands. It is another fact that concepts and properties in the classical domain ontology have crisp values.

However, most of the real-time systems hold the fuzzy terms [26, 32]. Consequently, the crisp ontology cannot ex-

actly define the knowledge of the system because of its limited expressivity function. The incorporation of fuzzy set theory with crisp ontology can yield a powerful knowledge base which can cover the ambiguity of the real-time domain. The fuzzy set theory was introduced by Lotfi Zadeh in 1965 [44] to deal with vague and imprecise concepts. To develop the fuzzy ontology, we divided the development process into two phases. At first, we developed the crisp domain ontology and evaluated the performance of the query using SPARQL, after that, we tagged the vague terms of the domain and by applying the *Fuzzy-OWL* plug-in of *Protégé* to incorporate the fuzzy terms in ontology [7, 8].

$$FDO = DE \times \{CO + FTE\} \quad (2)$$

where FDO is the fuzzy domain ontology, DE is the domain expert while the terms CO and FTE represent the crisp ontology and fuzzy term expression respectively. Ontology is widely acknowledged as a cornerstone in any knowledge base system. However, if we analyze the ontologies, we even find significant variation in the construction of some domain ontology. In literature, we can find several techniques for ontology development such as: *Tove*, *enterprise model approach*, *methontology*, etc. Unfortunately, all the said techniques are limited in scope and cannot fulfil the modern

information acquisition requirements. The ontology development techniques which we adopted, are so comprehensive, easy to modify, or extend and better capable to handle client queries [17]. We adopted the seven step approach which is mentioned in [31] to build our domain ontology at initial level. Even though, the ontology development steps are self descriptive yet the interested can find the more detail in [31]. The steps are listed below.

- Determine the domain and scope of the ontology
- Consider reusing existing ontologies
- Enumerate important terms in the ontology
- Define the classes and the class hierarchy
- Define the properties of classes
- Define the facet of the slots
- Create instances

After development of crisp domain ontology, the next stage is the evaluation of the ontology to test/evaluate the efficiency of the ontology. The protégé by default has several *Reasoner* tools like *FaCT ++* and *RACER*. These tools can automatically check the bindings of terms in the ontology and can further generate inference results on behalf of it. On completion of crisp domain ontology, the next step is the incorporation of fuzzy terms in the ontology [23, 24]. The Fig. 5 is highlighting the generic architecture of the

fuzzy scene independent ontology. The fuzzy ontology generation process can be divided into four stacks as mentioned in Fig. 5 with names: domain category stack, fuzzy class and relation stack, fuzzy class property stack and fuzzy set stack. The domain category stack holds all the fuzzy categories of the domain like sea depth, brightness level, etc. To get the most precise results, we manually tagged the fuzzy terms from domain data and exported the fuzzy terms to the *FUZZY-OWL* Plug-in of the *Protégé* to assign the fuzzy values. The second layer of Fig. 5 represents the fuzzy class and their relations, for example, in our test case we had AUV, obstacle, depth value, start and end positions.

We can define AUV and obstacle as classes while defining the relationship among them based on their characteristics. We incorporated the fuzzy set theory to represent the imprecise concepts in our pre-modeled ontology as the last step of fuzzy ontology development. We used *Fuzzy OWL* plug-in of Protégé which is a semi automatic tool for the creation of fuzzy ontology [7, 23]. This plug-in can help to create fuzzy data type, fuzzy concepts, fuzzy modifiers, and fuzzy axioms. Likewise, crisp ontology, the fuzzy ontology consists of classes, properties, instances and axioms, however, the fundamental difference between crisp and fuzzy ontology is that, all the concepts and most of their value are fuzzy terms in case of fuzzy ontology. Meanwhile, fuzzy

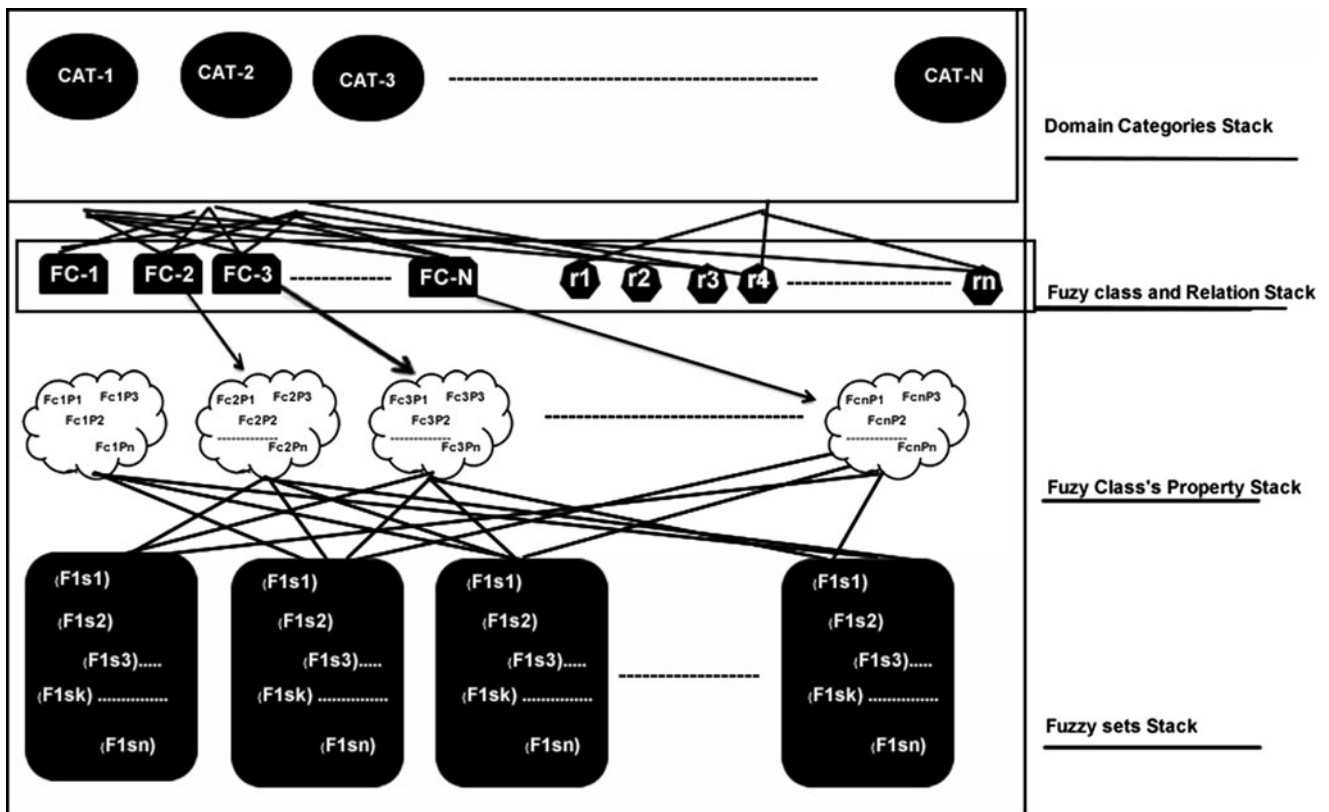


Fig. 5 Structure of fuzzy scene independent ontology

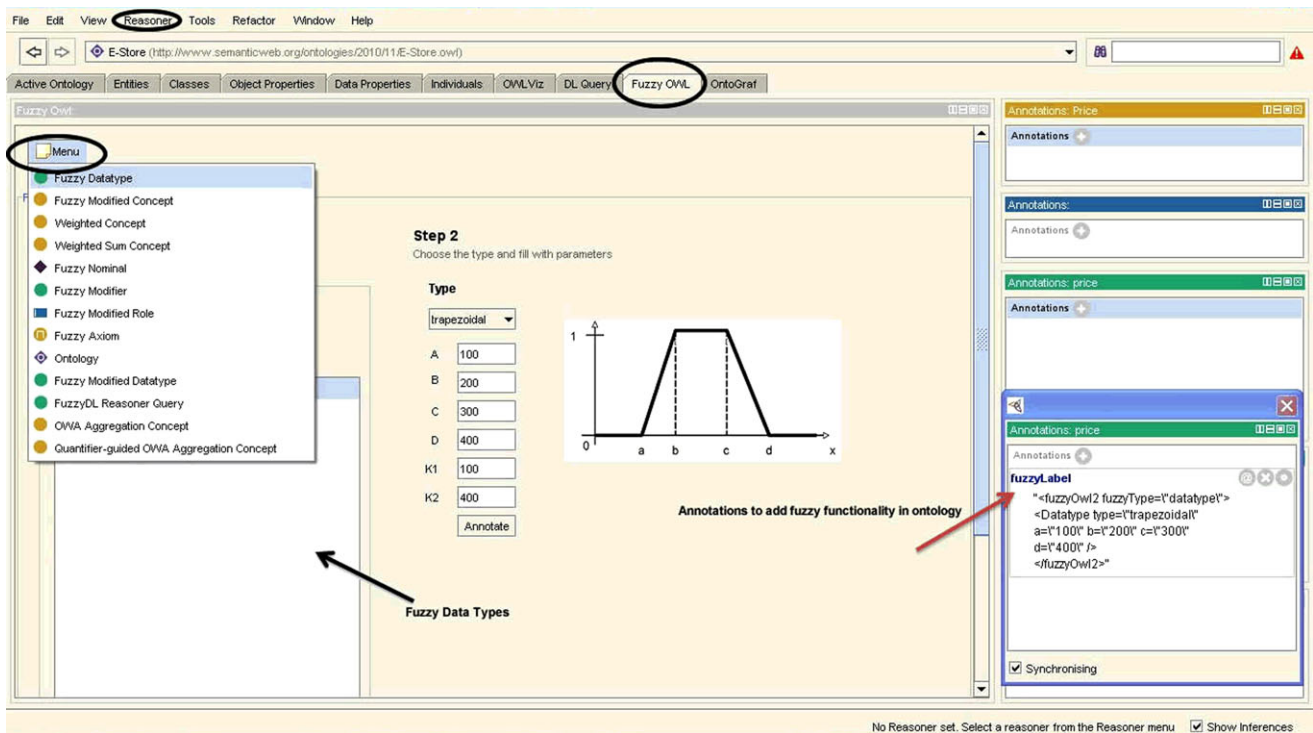


Fig. 6 Screenshot of fuzzy OWL plug-in of protégé

ontology can be defined in the form of fuzzy sets. Let $F\tilde{o}$ be a fuzzy class in the universe of discourse μ then $F\tilde{o}: \mu \rightarrow [0, 1]$ and the relationship between two ontology classes are fuzzy relation $A * B: \mu \rightarrow [0, 1]$. Figure 6 is a screen shot of Fuzzy OWL plug-in which is showing the featured menu, annotation pellet and *reasoner*.

The annotation feature of protégé is used to define a fuzzy concept in fuzzy ontology. The manual process of annotation adding is considered to be a complex and error pruning but the fuzzy OWL tab helps us to make this process handy by applying its automation power. We exported our domain ontology into fuzzy OWL tab by using the export feature of plug in, and added the fuzzy logic into already marked fuzzy terms. A class of sea water level can be described in a fuzzy format as: *SeaWater* \sqcap *hasLeve..Deep_level*. Similarly, *very deep* water level can be expressed as: *Very (Sea-Water)* \sqcap *hasLevel..Deep_level*. To confirm the efficiency of the ontology, several Reasoners are available such as: *DL reasoner*, *Pellet* and *DeLorean*. We applied the *DeLorean* reasoner to getting inference results from our ontology [13]. The internal working of *DeLorean* [6, 7] first converts the fuzzy ontology into crisp ontology then generates the inference results.

The linker and management ontology of the SKBL is designed to monitor all the information sharing tasks. It helps the operator in information extraction from multiple ontologies. The development process of the ontology is the same as defined above. As an addition, we just added the roles (job

description) of each component of SKBL in this ontology. It monitors the working of each part of SKBL and helps in resource sharing. The VR CAD model repository is the linked directory of VR model; we discussed this portion in Sect. 5.

3.1 Ontology evaluation

The basic purpose to evaluate the ontology is to get assurance that the ontology is presenting the domain correctly or not, we adopted two ways to evaluate the ontology. The first was the evaluation through competency questions. Competency question is one best way to judge the effectiveness of the ontology. We used Protégé DL-query plug-in to acquire from ontology. Table 1 is presenting the competency question, DL-queries and results.

At the second stage of ontology evaluation, we conducted a workshop and invited all the stakeholders (AI algorithm developer, marine engineer, and end user, 3D VR developer) to participate. We presented the holistic view of the ontology and expressed the way, how we formalized the marine operations in ontology, to get suggestions and to validate our design strategy.

4 Fuzzy scene independent ontology to virtual scene mapping layer

This layer works as a mediator between semantic knowledge base layer and semantic virtual reality graphics en-

Table 1 Ontology evaluation through DL-query

Competency question	DL-query	Query results
Which kind of robots can perform autonomous marine operations?	<i>Auto_Robot and asWorking only marine_oper</i>	AUV ROV Autonomous Ship
Who are the team members involved in marine operation?	<i>Role and startProcess value MarineOperation</i>	Marine engineer Algorithm designer 3D developer Tester

gine layer. The primary purpose of this layer is to map the classes and instances of ontology to VRML objects and VR scene. In SKBL, we have designed the VRML (virtual reality markup language) ontology. The VRML ontology holds all the modeling primitives of the virtual reality scene. The mapping process includes the object level mapping, scene level mapping and domain level mapping [41]. We used the NeOn toolkit API in the development of our prototype. The alignment plug-in of NeOn excellently maps the fuzzy marine ontology into the virtual reality scene [36] as displayed in the simulator interface Fig. 9.

When a client wants to develop a VR test bed, he/she enters the natural language query by clicking the VR scene builder utility of the simulator. The natural language processing function of the simulator accepts the input and converts the NLQ to formalized SPARQL query. Subsequently, the sophisticated algorithm of VR scene builder utility applies the query against fuzzy VR Scene ontology and extracts the results in the form of ontology triples. After that, that automatic mapping algorithm of the prototype intelligently maps the ontology triples to VRML nodes and display the results as scene graph objects. Figure 7 explains the steps from natural language input to 3D virtual reality scene graphically. Each object in the VR scene is associated with an ontology instance which is further linked with the semantic knowledge base. As the VR ontology is the generalized ontology for VRML therefore, it can handle the VRML constraints also.

5 Semantic virtual reality graphics engine layer

5.1 3D CAD system

Computer aided design (CAD) is the modeling of physical systems on computers, allowing both interactive and automatic analysis of design variants and the expression of designs in a suitable format for manufacturing. Mostly, VRML geometry models are used to set up the virtual environment and these complex geometry files hold the information about the shape and characteristics of the model. The

complexity increases, when all the geometry models along with the virtual environment information which comprises of many models assembled together. To reduce these complex computations, a strategy is rendered to compensate the loss of topological information during the translation process of models from CAD to VR systems, the mock-up models will be translated as Models and Environment files.

The models and environment files will be written in the XML format [29]. For the CAD environment, the product models, tools and fixture models are designed in a commercial CAD system such as Pro/Engineer or Solid Works because of the difference of data expression format between CAD systems and VR system. However, in VR environment the models are usually expressed only by polygons, which cannot be used for virtual assembly operation, planning and evaluation. An automatic data integration interface should be developed to finish data transformation from CAD to VR. By using the CAD API development kit, the related information and data can be extracted from CAD neutral files and CAD inner database. Four types of data are mainly taken into account, including geometry data, topology data, assembly data and physics data. These models and data are then input into a virtual reality environment [9].

For the virtual reality environment, three components are necessary to construct a keyboard and Joystick-based virtual assembly environment. After geometry data, topology data, assembly data and physics data are transformed from CAD to VR, a hierarchical data model is designed to represent and reorganize these models and data in a virtual environment, and then an efficient scene graph structure is also designed to display and manage virtual objects. At the same time, virtual factory, texture, light and the surrounding environment are also set up to simulate real production scenarios [16]. Primarily, the virtual reality simulation is to carry out maintenance and analysis operations and to give the user a pre-processing data and knowledge of the environment as in real life. The kernel of virtual reality simulation is a multi-thread method to realize: graphic display, collision detection, constraint recognition, physics computation and dynamic simulation.

“Draw an underwater 3D environment with **one AUV** which has **start position** (0.0, 0.0, 0.0), **translation** (0.0, 0.5, 0.0) and **end position** (12.0, 12.0, 0.0) and **obstacle_1** with position (5.0, 5.0, 0.0), **obstacle_2** with position (11.0, 5.0, -5.0), **obstacle_3** with position (9.0, 8.75, -2.0). Translate AUV from start to end position and avoid the obstacles.”

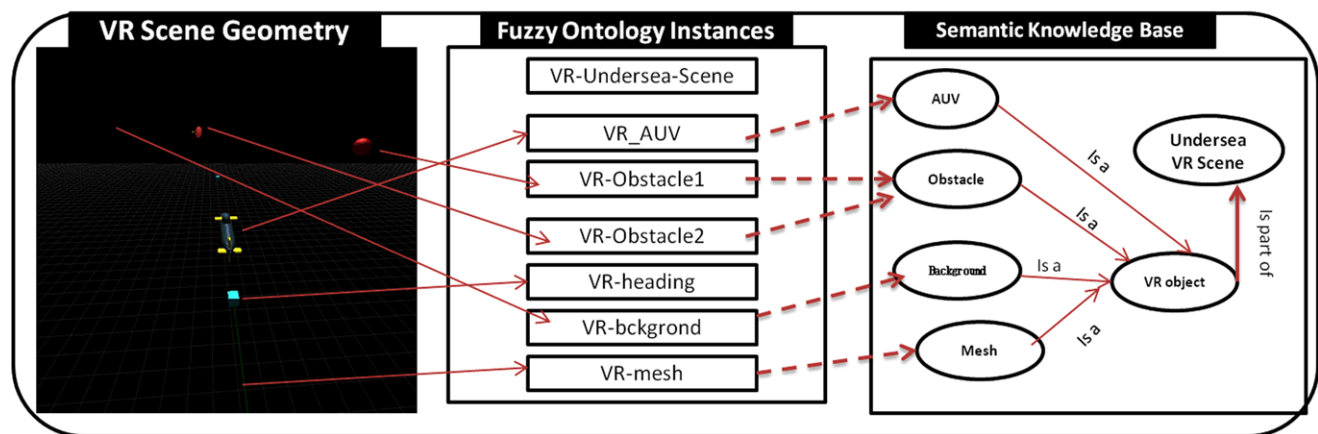
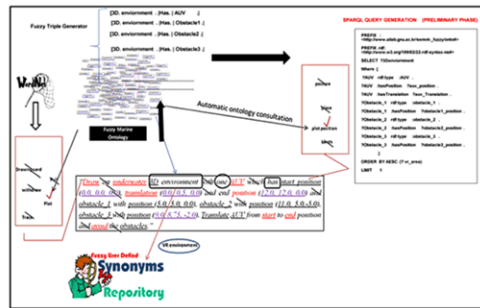


Fig. 7 The fuzzy ontology based semantic annotations of the objects

5.2 Model and environment files

Model files are proposed for each model with additional information about the joint configuration of the models. These files provide the information about the joint configurations at program load stage which is later used in calculating the real time kinematics simulations for marine robots. The model files are coded in XML format and the tags are named as joint names that will follow a hierarchical structure where by representing the needed information effectively. The nodes of the VRML model can be converted into frames of XML with additional information about the joints [48].

The attribute type identifies the joint either its prismatic or resolute followed by the axis attribute that defines the axis of rotation or movement of the joint. The model file further contains information like the name, ID of the element for visualization and the relative position and orientation of the feature in the element’s local coordinate system. The benefit of XML formatting is that its generic type can be used by any software and the information can be shared among var-

ious simulators with the help of XML feeds. Whereas the environment file is responsible to set up the layout of the virtual reality environment. We call all the VR models in the environment file and assign the position and rotation. The environment file forwards all the parameters to the semantic VR graphics engine which further plots the environment according to specification.

5.3 Hierarchical model & scene description

The hierarchical model is also called scene graph, which is a collection of nodes in a graph or tree structure. A node may have several child nodes but only have one parent node; an operation performed on a group automatically propagates its effect to all of its members. A scene description can be explained as a diagram that has a bunch of lines connecting a bunch of points. The points are the objects and transformations (called nodes) derived from the model files, like an AUV, or a Ship, or the ability to rotate something around the X-Axis by 45 degrees. The lines are an indication of the

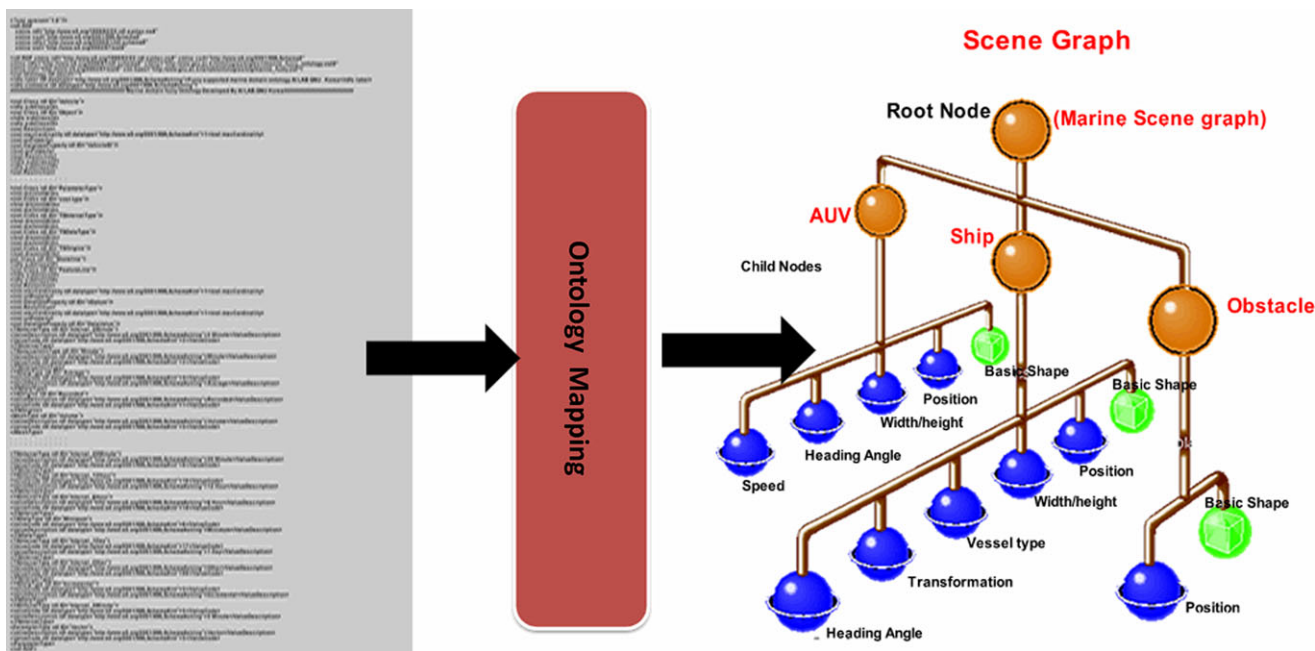


Fig. 8 Hierarchical architecture of semantic powered scene graph

relationship between the points [5, 18]. For example, a collection of simple objects can be grouped together to form a more complex object, in which case the node for the complex object called the *parent node* and the simple objects called *child nodes*. Figure 8 shows a sample scene graph hierarchical model (Scene description).

5.4 Application interface

The primary goal is to design desktop based application with flexible architecture supporting both fuzzy ontology and digital mockup architecture in order to present the exact virtual view of the physical scene. To control the features of this digital mock-up, the models XML based hierarchy is suggested. Prior to simulations, the model files from VRML (Virtual Reality Markup Language) models would be generated and environment files from the model files will be passed to the application to set up the whole environment. The software libraries to create the applications are proposed: Object oriented visual J++ as the programming language and Microsoft Visual Studio 2005 as the development environment. Object oriented programming concepts will be used to combine the functionality of each library.

The *Open Inventor*, a library for creating virtual environments, provides the application platform for this research. The *Open Inventor* API hides many of the low-level programming details required to develop, test, and debug VR applications unlike OpenGL [15]. After loading the VRML models, the application triggers two threads to handle the mission algorithm and to manage the graphical view. Open

inventor and VRML ontology are used to launch the graphics thread; depending on the performance of the desktop system, it can render the entire graphics scene as shown in Fig. 9.

6 Experiments and results

To evaluate the efficiency of the proposed scheme, we developed a Java based prototype. The reason to choose Java as development language is its robustness and platform independence attributes. Moreover, we integrated several open sources APIs and tool kits such as: VRML, Open Inventor, Protégé, NeOn ontology mapping plug-in and XML Meta Mapping to build diverse software. What is more, our experimental environment had Intel (R) Core (TM1) 2 Quad CPU with 4.0 GB RAM, and Window XP as operating system. For comparative analysis purpose, we developed the simulator on flexible bases which means it can easily be switched from non-ontology mode to crisp ontology mode and further from crisp ontology mode to fuzzy ontology mode and vice versa. In the non-ontological mode, the software behaves as a simple programming IDE and the users have to perform all the tasks through manual work or with programming, the conventional simulators which are usually developed in VC++, Java can be the examples of non-ontology mode.

In the crisp ontology mode of the software, we integrated the crisp ontology utilities with a conventional simulator; this amalgamation generates the automation capabilities in the simulator at certain extent. Last but not the least, the

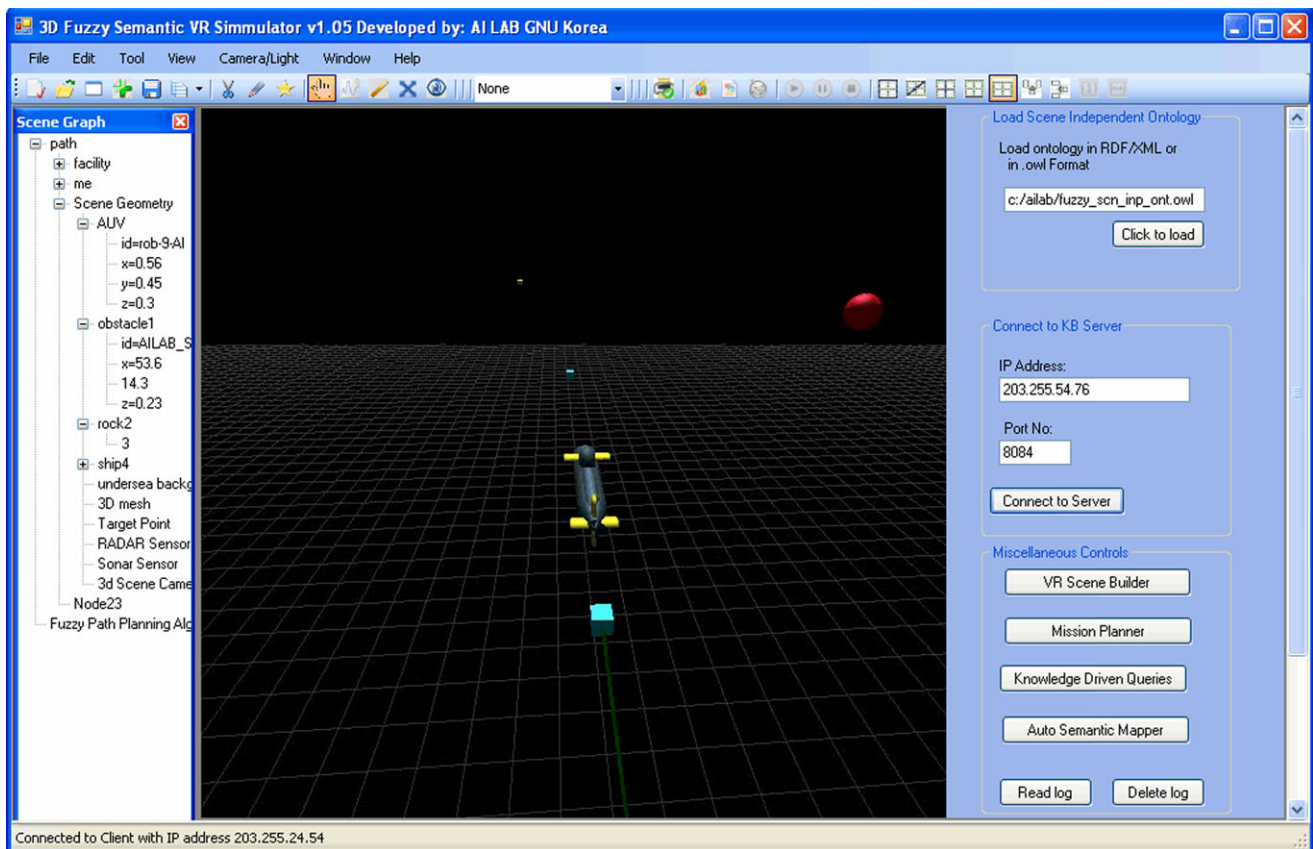


Fig. 9 Screenshot of prototype 3D AUV simulation

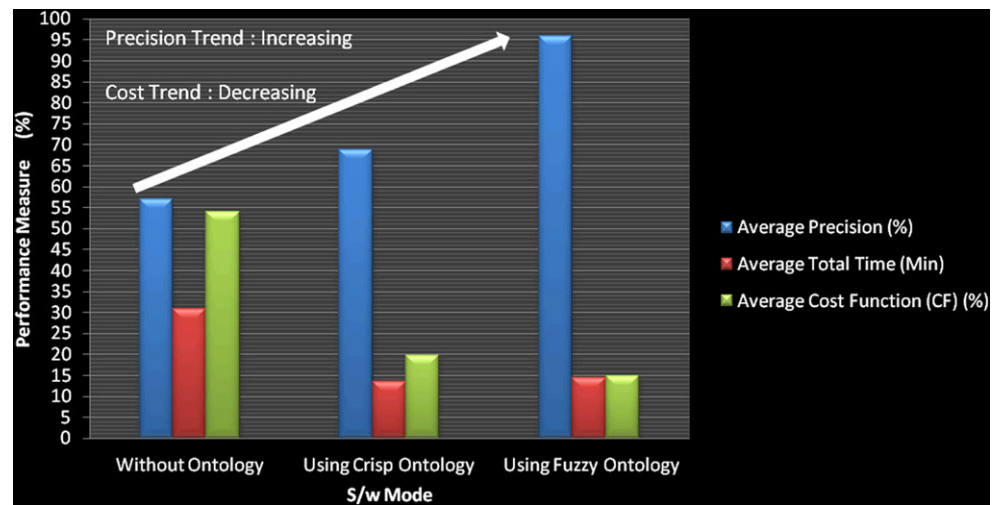
fuzzy ontology mode which comprises of fuzzy ontology and NLP, not only completely automates the system but it also fixes the high cost and precision issues related to the crisp ontology mode. We designed a real-time scenario with the help of marine engineers to verify the productivity of the proposed system. According to scenario, we had an intelligent algorithm, which we already had developed using the fuzzy BK—product to detect and avoid the underwater obstacles while gathering the environment information [43]. Our developed algorithm automatically acquires the input values from external sensors like sonar and other motion sensors to avoid the underwater obstacles. If we had to monitor the effectiveness of our algorithm; we would have to utilize the underwater robot equipped with multiple sensors.

Apparently, the real-time experiments demand for high cost and skilled team. As an alternate, we performed this experiment by using our proposed simulator. A client can easily draw a 3D virtual reality scene which mimics the real-time environment by utilizing VR scene building utility of the simulator. The obstacle avoidance and path planning algorithm can be integrated with the 3D system while exchanging some parameters. During the experimental phase, we requested five volunteers (A, B, C, D and E) to help us in order to analyze the performance of the overall system. All

the five users had been selected based on their level of expertise. The first user was the expert user and had complete information about the virtual scene developing and marine domain, the second was the mid level user who had medium level knowledge about underwater water experiments and algorithm working, the third user was an AI algorithm developer nevertheless he had very low knowledge about 3D graphics designing and virtual testing. Similarly, the fourth and fifth were basic and non-technical user respectively and they did not have much knowledge about the virtual experiment. Regardless, they had some 3D game playing experience in a virtual environment.

Before starting the experiments, we provided a brief overview of the system to the testers and handed over user manual to operate the software. The user manual had the information about the basic 3D environment designing parameters; therefore even a non-technical user can understand the basics of the system. During the experiments, we guided the evaluators to use each mode one by one in order to carry out the experiments. Additionally, we monitored the total execution time, which each user had spent to perform a scenario by using each mode. The VR scene builder and mission planner are based on NLP and fuzzy ontology. Therefore, we cannot ignore the aspect of information extraction,

Fig. 10 The precision and cost comparison graph between non-ontology, crisp ontology and fuzzy ontology with NLP modes



as the entire scene building process is highly dependent on the extracted information through ontology, drawn with natural language query. An information system can be categorized due to its preciseness and usability. There are a few known measures through which we can analyze the productivity of any information system.

The precision and recall are the parameters which can guide us to analyze the performance of the system accurately. Here, we introduced a cost function (CF) to monitor the overall performance of the system. By using the following formulae, we can calculate the precision, recall and cost function of any information system.

$$\text{Precision (PR)} = \frac{A_r}{A_r + F_r} \quad (3)$$

$$\text{Recall (RL)} = \frac{A_r}{A_r + T_r} \quad (4)$$

$$\text{Cost Function (CF)} = \frac{\text{Execution Time (ET)}}{\text{Precision (PR)}} \quad (5)$$

where A_r is total number of the resources which are extracted with natural language query and T_r , F_r are the true and false resources respectively. Table 2 is displaying the complete list of the true and false sources extracted during the experiments while applying all the modes of the software. For analysis purposes, we calculated the average precision and recall rates on the completion of query processing activity. Figure 10 clearly demonstrates the significant increase in precision by using fuzzy ontology. The observed precision was 57.06 % in the case of non-ontology mode which increases to 95.78 % when we applied the fuzzy ontology. These precision values evidently validate the efficiency of the fuzzy ontology. On the other hand, the cost function is the ratio of total execution time to the precise information which is extracted. The cost function includes the time which is taken in manual editing of the scene in case of crisp ontology mode. With the help of cost function, we

can accurately measure the total cost of the system. In our system, we observed a drastic slump in the overall cost of the system from 53.9 % to 14.9 % by using non-ontology to fuzzy ontology respectively.

If we critically analyze the presented statistics of Table 2, we can find that the total time elapsed in case of fuzzy ontology and NLP mode is slightly greater than the total time elapsed in crisp ontology mode. The extra processing time is considered the major reason behind this delay, the fuzzy ontology and NLP mode is the merger of the three sophisticated technologies and the processor have to perform extra computation when we execute a query using fuzzy ontology mode. The extra computation means the increase in total execution time. However, this is interesting to note that the delay factor could not affect the overall performance of this system as in crisp ontology mode, the system can only acquire the crisp available information and its infeasible to draw the desired scene with the help of information extracted in case of crisp ontology. Consequently, the users have to manually edit the scene which certainly takes a too much time.

Further, a lot of information is available in the form of imprecise concepts which could be extracted by using fuzzy ontology. The fuzzy ontology mode empowers the system to get the complete information about the scene by using the fuzzy ontology reasoning engine; therefore, the system quickly manipulates the information and draws the scene according to user's requirements without user manual contribution. We have analyzed that it took more time to execute the whole process using crisp ontology rather using fuzzy ontology. More simply, we can say that in the crisp ontology mode, the algorithm only extracts the crisp information from the knowledge base so it consumes less time. On the other hand, in fuzzy ontology and NLP mode, the algorithms tries to find the hidden information using reasoning capability of the fuzzy ontology. Therefore, the query execution time increases. In addition to this, we can observe some

Table 2 Statistical performance analysis of proposed system

	Resource extracted by query (R)	Extracted true elements (T)	Extracted false elements (F)	Recall (%)	Precision (%)	Total execution time (min)	Cost function (%)
Without Ontology							
A (Expert User)					97.3	15.7	16.2
B (Mid Level User)					72.0	22.5	31.2
C (AI Developer)					79.0	29.4	37.2
D (Basic User)					27.0	38.6	14.3
E (Non-Tech. User)					10.0	47.2	47.3
Average					57.06	30.8	53.9
Using Crisp Ontology							
A (Expert User)	419	300	119	58.3	77.9	5.2	6.50
B (Mid Level User)	500	237	263	67.9	65.6	8.4	12.8
C (AI Developer)	433	284	149	58.5	74.4	13.7	18.4
D (Basic User)	512	202	310	71.8	62.3	17.6	28.2
E (Non-Tech. User)	456	175	281	72.3	61.9	23.1	37.1
Average					68.82	13.6	19.7
Using Fuzzy Ontology & NLP							
A (Expert User)	500	489	11	50.5	97.8	6.3	6.40
B (Mid Level User)	456	440	16	50.8	96.6	8.7	9.00
C (AI Developer)	512	490	22	52.0	95.8	14.5	15.2
D (Basic User)	419	400	19	51.2	95.6	18.1	18.9
E (Non-Tech. User)	433	402	17	51.9	93.1	24.2	25.9
Average					95.78	14.36	14.9

empty columns in Table 2 which are due non-ontology mode of the software.

We did not use the ontology technique and natural language processing feature in non-ontology mode of the software. Hence, we cannot observe the extracted true and false elements; nevertheless, we can calculate the precision ratio of output. In non-ontology mode, the precision of the experiment is based on the expertise of the operator which means if the operator is the expert user, he/she can get the best results compared with the novice user. During the experiments, we also have noted that by using a fuzzy ontology technique, a beginner or non-technical user can achieve the reasonable results which prove the effectiveness of our proposed system which could not be possible with non-ontology mode. Another significant feature of the proposed system is that we can modify or change the virtual test bed at any stage of the experiment which seems to be infeasible in case of non-ontology mode. In the non-ontology and crisp ontology mode, a slight change in the design requires a lot of programming, hence, the cost of the overall system increases.

7 Concluding remarks and future work

This research presents the idea of 3D fuzzy semantic supported virtual reality simulator to expedite the evaluation process of control algorithms in marine robots. The digital mock-up technology creates a virtual test bed to analyze the marine operations remotely. The incorporation of fuzzy domain ontology with 3D graphical engine facilitates the operator to create the virtual world through interactive utilities without knowing the complex underlying graphics programming. The intelligent architecture of the simulator is based on fuzzy domain ontology and NLP, the powerful fuzzy logic helps to cover up all the manifestation of real-time scenario in virtual worlds and to facilitate the user to acquire from the live virtual scene through NLQ. As the simulator architecture is for multipurpose so it can cover all kinds of experiments related to marine domain. The experimental data generated from different experiments, prove that the proposed simulator has the potential to become a valuable tool in marine robots rapid development. The automatic NLQ assisted virtual scene plotting utility of this simulator encourages the researchers from different backgrounds to design more sophisticated intelligent algorithms. We are working to formulate this tool more result oriented. As a future work, we will automate the fuzzy semantic knowledge base layer and will try to integrate the vocal utility to plan and execute the entire mission through the voice command interface.

Acknowledgements This work was supported by the Korea National Research Foundation (NRF) Grant funded by the Korean Government (No. 2010-0002346).

References

1. Anvar AM (2003) Intelligent navigation process for autonomous underwater vehicles (AUVs) using time-based fuzzy temporal reasoning. In: Proceedings of 10th international symposium on temporal representation and reasoning and fourth international conference on temporal logic, pp 56–61
2. Ahmad C, Kim Y (2011) Ontology-assisted automatic precise information extractor for visually impaired inhabitants. *Artif Intel Rev*, 1–16
3. Ahmad C, Kim Y (2011) Incorporation of fuzzy theory with heavyweight ontology and its application on vague information retrieval for decision making. *Int J Fuzzy Log Intell Syst* 11(3):135–216
4. Ahmad C, Kim Y (2011) Exploiting the heavyweight ontology with multi-agent system using vocal command system: a case study on E-Mall. *Int J Adv Comp Technol* 3(6):233–241
5. Baumeister J, Reutelshoefer J, Puppe F (2011) Know we: a semantic wiki for knowledge engineering. *Appl Intell* 3(25):323–344
6. Bukhari AC, Kim YG (2012) Integration of a secure type-2 fuzzy ontology with a multi-agent platform: a proposal to automate the personalized flight ticket booking domain. *Inf Sci* 198:24–47
7. Bobillo F, Delgado M, Gómez J, DeLoreau (2012) A reasoner for fuzzy OWL 2. *Expert Syst Appl* 39(1):258–272
8. Bobillo F, Straccia U (2011) Fuzzy ontology representation using OWL 2. *Int J Approx Reason* 52(7):1073–1094
9. Bram P, Wesley B, Olga DT, Frederic K (2005) VR-WISE: a conceptual modelling approach for virtual environments. In: Proceedings of the first international methods and tools for virtual reality (MeTo-VR 2005) workshop. Ghent, Belgium
10. Boumechaal H, Boufaïda Z (2011) Formalization of natural language queries. In: Proceeding of 2011 international symposium on innovations in intelligent systems and applications (INISTA), pp 495–499
11. Beji S, Kadhi N (2009) Security ontology proposal for mobile applications. In: Proceedings of tenth international conference on mobile data management: systems, services and middleware, pp 580–587
12. Brutzman DP, Kanayama Y, Zyda MJ (1992) Integrated simulation for rapid development of autonomous underwater vehicles. In: Proceedings of the 1992 symposium on autonomous underwater vehicle technology AUV '92, pp 3–10
13. Carloni O, Leclère M, Mugnier M (2009) Introducing reasoning into an industrial knowledge management tool. *Appl Intell* 3(31):211–224
14. Chong W, Miao X, Zhou Q, Yong Y (2007) PANTO: a portable natural language interface to ontologies. In: Proceedings of the 4th European conference on the semantic web: research and applications, pp 473–487
15. Chen Z (2010) Development of OpenGL based 3D simulator for computer numerical control. In: Proceeding of international conference on artificial intelligence and computational intelligence (AICI), pp 319–321
16. Christand YJ (2011) Assembly simulations in virtual environments with optimized haptic path and sequence. *Robot Comput Integr Manuf* 27(2):306–317
17. Dean J, Trevor B, Pepijn V (1998) Methodologies for ontology development, pp 62–75
18. Dietrich A, Wald I, Wagner M, Slusallek P (2004) VRML scene graphs on an interactive ray tracing engine. In: Proceedings of IEEE virtual reality, pp 109–282
19. Erric P, Andy S (2005) SPARQL query language for RDF. 2008. <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>
20. Francisco G, Barber O, Miguel L (2006) An ontology-based approach for ive+va. In: International conference on intelligent virtual environments and virtual agents (IVEVA). Aguascalientes, Mexico, pp 67–76

21. Gao H, Zhao J, Yin Q, Wang J (2009) Ontology-based enterprise information retrieval model. In: IEEE international conference on grey systems and intelligent services, GSIS 2009, pp 1326–1330
22. Javier A, Ortiz A (2005) Development of the control architecture of an underwater cable tracker. Department of Mathematics and Computer Science, University of the Balearic Islands, v 20, pp 477–498
23. Jesús IM, Antonio FS, Josep BG (2006) Fuzzy approach to the intelligent management of virtual spaces. *IEEE Trans Syst Man Cybern, Part B, Cybern* 36(3):494–508
24. Jesús IM, Carlos DM (2006) A basic semantic common level for virtual environments. *Int J Virtual Real* 5(3):25–32
25. Kobayashi E, Aoki T, Maeda T, Hirakata K, Ichikawa T, Saitou T, Miyamoto S, Iwasaki S, Kobayashi H (2001) Development of an autonomous underwater vehicle maneuvering simulator (OCEANS). In: MTS/IEEE conference and exhibition, 2001, pp 361–368
26. Lee CS, Wang MH (2009) Ontology-based computational intelligent multi-agent and its application to CMMI assessment. *Appl Intell* 3(30):203–219
27. Li W, Chen G, Kong Q, Wang Z, Qian C (2011) A VR-Ocean system for interactive geospatial analysis and 4D visualization of the marine environment around Antarctica. *Comput Geosci* 37(11):1743–1751
28. Macedonia MR, Brutzman DP, Zyda MJ, Pratt DR, Barham PT, Falby John L (1995) NPSNET: a multi-player 3D virtual environment over the Internet. In: Proceedings of the 1995 symposium on interactive 3D graphics, Monterey, California, United States, pp 93–99
29. Moorthy S (1999) Integrating the CAD model with dynamic simulation: simulation data exchange. In: Proceedings of the 1999 winter simulation conference, vol 1, pp 276–280
30. Nasrolahi S, Nikdast M, Boroujerdi MM (2009) The semantic web: a new approach for future world wide web. In: World academy of science, engineering and technology
31. Noy NF, McGuinness DL (2001) Ontology development 101: a guide to creating your first ontology. Accessed 2011. <http://protege.stanford.edu/publications/ontology-development/ontology101-noy-mcguinness.html>
32. Paar A, Reuter J, Soldatos J, Stamatis K, Polymenakos L (2009) A formally specified ontology management API as a registry for ubiquitous computing systems. *Appl Intell* 1(30):37–46
33. Ridao P, Carreras M, Ribas D, El-Fakdi A (2004) Graphical simulators for AUV development. In: First international symposium on control, communications and signal processing, pp 553–556
34. Segev A, Kantola J (2010) Patent search decision support service. In: Seventh international conference on information technology: new generations (ITNG), April 12–14, 2010, pp 568–573
35. Shadbolt N, Hall W, Lee T (2006) The semantic web revisited. *J Intell Syst* 21(3):96–101
36. Suarez-Figueroa MC, Gomez-Perez A (2009) NeOn methodology for building ontology networks: a scenario-based methodology. In: Proceedings of the international conference on software, services and semantic technologies (S3T 2009), pp 146–156
37. Tho QT, Hui SC, Fong ACM, Cao TH (2006) Automatic fuzzy ontology generation for semantic web. *IEEE Trans Knowl Data Eng* 18(6):842–856
38. Valencia R, García F, Castellanos D, Fernández T (2011) OWL-Path: an OWL ontology-guided query editor. *IEEE Trans Syst Man Cybern, Part A, Syst Hum* 41(1):121–136
39. Xiuwen L, Cui X, Yicheng J (2009) Multi-level virtual reality system for marine education and training. In: First international workshop on education technology and computer science ETCS '09, vol 2, pp 1047–1050
40. Yan P, Zhao Y, Sanxing C (2008) Ontology-based information content security analysis. In: Proceedings of fifth international conference on fuzzy systems and knowledge discovery, pp 479–483
41. Yang K, Steele R (2009) Ontology mapping based on concept classification. In: 3rd IEEE international conference on digital ecosystems and technologies DEST '09, pp 656–661
42. Yi S, Dezheng Zh, Li Ch (2010) Fuzzy ontology constructing and its application in traditional Chinese medicine. In: IEEE international conference on intelligent computing and intelligent systems (ICIS), vol 2, pp 592–595
43. Kim Y-G (2009) An autonomous navigation system for unmanned underwater vehicle, underwater vehicles, pp 279–294
44. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8(3):338–353
45. Zhai J, Lv M, Li J, Wang Q (2008) Knowledge modeling for fuzzy systems based on ontology. In: Control and decision conference, CCDC 2008, pp 3393–3397 (Chinese)
46. Zhai J, Shen L, Liang Y, Jiang J (2008) Application of fuzzy ontology to information retrieval for electronic commerce. In: Proceedings: international symposium on electronic commerce and security, pp 221–225
47. Zhai J, Wang Q, Lv M (2008) Application of fuzzy ontology framework to information retrieval for SCM. In: International symposiums on information processing (ISIP), pp 173–177
48. Zhang S, Song L, Niu L (2008) Study on X-VRML-based modeling of virtual museum. In: Eighth international conference on intelligent systems design and applications, ISDA '08, vol 3, pp 285–288
49. Zyda MJ, McGhee RB, Kwak S, Nordman DB, Rogers RC, Marco D (1990) Three-dimensional visualization of mission planning and control for the NPS autonomous underwater vehicle. *IEEE J Ocean Eng* 15(3):217–221



Ahmad C. Bukhari received his first master degree in information technology from PUCIT Lahore in 2008 and second master degree from Gyeongsang National University Korea in 2012. His area of research includes, semantic web, soft computing, intelligent system, Artificial Intelligence. He has couple of research articles to his credit.



Yong-Gi Kim received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea in 1978, the M.S. degree in computer science from University of Montana, USA in 1987, and the Ph.D. degrees in computer and information sciences from Florida State University, USA in 1991. He was a Visiting Scholar in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Illinois, USA, from 2008 to 2009. He is currently a Professor in the Department of Computer Science, Gyeongsang National University, Korea. His current research interests include soft computing, intelligent systems and autonomous underwater vehicles.