A Semantic Management Method of Simulation Models in GNSS Distributed Simulation Environment

Guo-chao FAN^{†a)}, Chun-sheng HU^{††}, Nonmembers, Xue-en ZHENG^{†††}, Member, and Cheng-dong XU^{†††}, Nonmember

SUMMARY In GNSS (Global Navigation Satellite System) Distributed Simulation Environment (GDSE), the simulation task could be designed with the sharing models on the Internet. However, too much information and relation of model need to be managed in GDSE. Especially if there is a large quantity of sharing models, the model retrieval would be an extremely complex project. For meeting management demand of GDSE and improving the model retrieval efficiency, the characteristics of service simulation model are analysed firstly. A semantic management method of simulation model is proposed, and a model management architecture is designed. Compared with traditional retrieval way, it takes less retrieval time and has a higher accuracy result. The simulation results show that retrieval in the semantic management module has a good ability on understanding user needs, and helps user obtain appropriate model rapidly. It improves the efficiency of simulation tasks design.

key words: GNSS, distributed simulation system, model management, resource description framework, semantic web

1. Introduction

PAPER

The GNSS (Global Navigation Satellite System) Distributed Simulation Environment (GDSE) is a simulation environment oriented Internet, which is proposed according to the sharing ideas of generalized cloud computing [1]-[6] and Service Oriented Architecture (SOA) [7]-[9]. The simulation environment constructed in this way has a strong sharing ability and reusability in theory, which will cut the cost of development and improve the capability of the simulation environment [3], [10]. Because of a large amount of models in GDSE, models retrieval might be an extremely complex project, and therefore simulation task design [11]. Meanwhile, the service-oriented simulation model has various special property and relation to describe, which can't be satisfied with traditional management method. For example, the incidence relation among models can't be described well in Database; and the relation between combination model and the model ontology also can't be represent clearly in Database. There also may be a retrieval operation for each model, i.e. the retrieval times are equal to the quantity of

Manuscript revised August 1, 2018.

Manuscript publicized October 9, 2018.

[†]The author is with the Beijing Jinghang Computation and Communication Research Institute, Beijing, China.

^{††}The author is with the School of Mechanical Engineerning, Ningxia University, Yinchuan, Ningxia, China.

^{†††}The authors are with the School of Aerospace Engineerning, Beijing Institute of Technology University, Beijing, China.

a) E-mail: fanguochao@yeah.net

DOI: 10.1587/transinf.2017EDP7377

models needed by user. Hu Chunsheng initially proposed a models management method for distributed simulation system in his papers [12]–[14]. The method is used for recommending next model which may be useful to users, when a model has been selected. It is a method to decrease retrievals, but may not be needed by users because a model may be satisfied for a variety of simulation needs. So the method is not an effective solution way for retrieving the appropriate model rapidly. Semantic web provides a new retrieval way on the basis of user's requirement [15], [16]. As the resources description rules of semantic web, RDF (Resource Description Framework) and OWL (Web Ontology Language) provide the basic ontology descript method [17]-[20]. Generally, this way may give all necessary models in one retrieval, which avoids the frequent retrieval operation. It will cost less retrievals for the same group models than traditional retrieval method, and therefore improves user retrieval efficiency. Management objects of Semantic Web are usually the completely independent entity, such as books, merchandise, etc. [21], [22]. However, the simulation models have many relations with each other. So a simulation model-specific semantic management method is necessary to retrieve the appropriate model rapidly.

Based on the study of RDF and OWL [17], [18], [23], a semantic management method of the simulation model is proposed in this paper. Firstly, Some main characteristics of service-oriented simulation models are analyzed in four cases, i.e. model ontology, combination model, incidence relation and simulation task flow. Then the resource description format and content of service-oriented simulation model are putting forward respectively. Based on the abovementioned research, the service-oriented simulation model management module of GDSE is designed. The framework of the module and the relationship of description files are given for realizing the semantic management method of service-oriented simulation model. Compared with the simulation results of traditional method, the result shows that retrieval through requirement (semantic management method) has a good ability to understand user needs, and helps user obtain all appropriate models rapidly in the least retrievals. Therefore, it has ability to reduce the total retrieval time.

Manuscript received November 22, 2017.

2. Analysis of Service-Oriented Simulation Model in GDSE

2.1 The Property of Model Ontology

All models need to be encapsulated with service before used in GDSE. The models with service-oriented encapsulation have a unified interface, and is noted as service-oriented simulation model. Calling rules and parameter configuration of service-oriented simulation model should be met when the model is called. In this way, there is a unique calling method for models that programed with different programming language (such as C, C++, C#, Matlab, etc.). It provides an implementation way to manage models with interface and registry information.

Each model provides simulation service with its particular URI (Uniform Resource Identifier), which is used as ontology identifier. There are many types of properties and parameters of service-oriented simulation model, including the research field, basic property, input data, output data, etc. Each property or parameter also has a value. Some value is numeric, and some are other resources. The above characters just adapt the RDF rules (Triple Property), so that service-oriented simulation model can be described with RDF rules.

A description of service model "A" is defined as follows.

$$M_A = S (SYS_A, P_A, Init_A, IN_A, OUT_A)$$
(1)

Where SYS_A is the navigation system information set of model A, whose value is one or more GNSS systems, such as GPS, BDS, etc.

 P_A is the basic property of model A. It consists of some basic information, including name, description, etc. So P_A is defined as Eq. (2).

$$P_A = \{Name, description, \cdots\}$$
(2)

 IN_A and OUT_A are respectively input parameters and output parameters. There may be not one parameter (input or output) in a model. And the parameters (input or output) may have a great difference between different models. IN_A and OUT_A are non-fixed-size sets. Size of set is determined by the number of input parameters or output parameters.

$$IN_A = \{ p_1^{In}, p_2^{In}, \cdots \}$$
(3)

$$OUT_A = \left\{ p_1^{Out}, p_2^{Out}, \cdots \right\} \tag{4}$$

 $Init_A$ is the initialization information of model. Each element in $Init_A$ has to be set a value while calling the model.

2.2 The Configurability and Composability

Unlike most resources on the Internet, the service-oriented simulation model not only has Triple Property, but also has configurability and composability.

Configurability: In order to run the simulation task, the service-oriented simulation model need to be configured during task design on GDSE. There may be various differences among models. To understand the model configuration before calling model, it should be described when register to GDSE.

Composability: Combined with each other, some service-oriented simulation model constructs the simulation task. So service-oriented simulation model has composability for combination. The model combination is defined as

$$M_C = Co\left(M_A, M_B\right) \tag{5}$$

Where model C is the combination of model A and B. Co() represents the combined operation of models. It shows a relation from model A to model B. After combined, model C is still shown as a model. So it can be noted as

$$M_C = S \left(S Y S_C, P_C, Init_C, I N_C, O U T_C \right)$$
(6)

However, the combination model has a series of differences from non-combined model, such as simulation function, system, etc. It should be shown in the description. For example, the navigation system information set should be the system available for each model in combination. The system of combination model *C* is the intersection of the system of model *A* and model *B*; the property is the union of the property of model *A* and model *B*, So Co() is defined with a group operation as follows.

$$SYS_C = SYS_A \cap SYS_B \tag{7}$$

$$P_C = P_A \cup P_B \cup P_C = Sub_C \cup P_C \tag{8}$$

$$Init_C = Init_A \cup Init_B \tag{9}$$

$$IN_C = IN_A \cup (IN_B - OUT_A) \tag{10}$$

$$OUT_C = OUT_B \cup (OUT_A - IN_B) \tag{11}$$

Where Sub_C is the basic information of all sub-model in *C*, such as M_A , M_B , etc. \hat{P}_C is the new properties after combination.

2.3 The Incidence Relation

Each service-oriented simulation model is just an independent unit in GDSE. Based on its own simulation ability, service-oriented simulation models construct the simulation task with each other through interfaces (input interface, output interface). During the combination, some relations are built among models, which can be noted as the incidence relation. The relation between two models which connected directly is called direct incidence relation. While the relation that didn't connect directly is called indirect incidence relation.

There may be many incidence relation between one service-oriented simulation model with others, while each relation degree is different from others. Because some models are usually used together, and some may be not. Some models even can't be used together. So it is needed to define an index for describing the incidence relation. The index is defined as the model incidence relation degree (MIRD). It is noted as $d(M_A, M_B)$ and represents the MIRD from model *A* to model *B*.

2.4 Simulation Task Flow

After combination with each other by incidence relation, service-oriented simulation models formed a set of models. It is noted as the simulation task flow, which has a particular simulation function. Figure 1 shows a simulation task flow for generating pseudorange in GNSS simulation system. It includes User-Track Model, WGS-84 Model, Rinex-Ephem Model, Satellite-Position Calculation Model, Satellite Clock Error Model, Klobuchar Ionospheric Delay Model, Hopfield Tropospheric Delay model, etc. The incidence relations are shown as the arrows in the figure.

In Fig. 1, User-Track produces the user's position with latitude, longitude and height (B, L, H) which were transform to XYZ coordinate user position (x_u, y_u, z_u) in WGS-84 coordinate by BLH-XYZ model. With the parameters (t_{oe}, i_0, \cdots) from Rinex-Ephem model, the satellite position are calculated into XYZ coordinate position $(x^{(s)}, y^{(s)}, z^{(s)})$ of WGS-84 in each simulation time epoch (t). The visibility of satellite is judged through Range & Visible Judgement model according to user position (x_u, y_u, z_u) , satellite position $(x^{(s)}, y^{(s)}, z^{(s)})$ and shielding angle (θ_u) . Simultaneously, the range (r) between user and satellite is calculated by Range & Visible Judgement model. Then pseudorange is generated with range(r), satellite clock error ($\delta t^{(s)}$, result of Satellite Clock Error model), ionospheric delay (I, result of Ionospheric Delay model) and tropospheric delay (T,result of Tropospheric Delay model). The parameters transformed between models construct the model incidence relation. The whole combination in Fig. 1 has a simulation function to generate pseudorange with simulation time.

Besides the above information, a completed simulation task flow also includes simulation task flow property, service-oriented simulation model set, incidence relation setting, flow driving, etc. The simulation flow property includes task name, description, step, etc. The serviceoriented simulation model set is made up of all models in simulation task flow. Incidence relation setting is the output, input and initialization information among models. The



Fig. 1 Single pseudorange simulation task flow

flow drive is the basic information and operation to call models.

The definition of simulation task flow is

$$T_{test} = Co(M_A, M_B, \cdots) + cfg_{test}$$
(12)

 $Co(M_A, M_B, \cdots)$ is the combination information of all model in simulation task flow. And cfg_{test} is the basic configure information of simulation task flow. And $cfg_{test} = \{Dr_{test}, P_{test}\}$. The Dr_{test} is just the flow drive and P_{test} is some more properties of flow, such as flow property, service-oriented simulation model set, incidence relation setting, etc.

3. The Resource Description Design for Service-Oriented Simulation Model

3.1 The Model Ontology Description

Based on the analysis in Sect. 2, service-oriented simulation models possess four characteristics, namely triple property, configurability, composability and model incidence relation in GDSE.

As a resource description object, the described content of service-oriented simulation model includes name, introduction, creator, publisher, version, navigation system, code language, input parameters, output parameters, initialization parameters, etc. The input parameters, output parameters and initialization parameters need to be defined separately. The above triple property set is entirely describing the simulation model with RDF rule. The basic RDF format is presented as Fig. 2. And the resources description file example of model ontology is shown as follows.

As the definition of model ontology, M has five properties, namely navigation systems, the basic properties of the model, input parameters, output parameters and initialization parameters (Fig. 2). The same type simulation models



Fig. 2 The schematic diagram of simulation model resource description



Fig. 3 The combination model description

may have significant differences in different navigation systems. Therefore, the model needs to indicate the applicable navigation system in the simulation model registration. The basic properties of the model are mainly used for describing the models' own properties, such as name, version, model descriptions, and so on. It increases the semantic reasoning ability of management module, which helps users with more accurate searching result. Input parameters, output parameters and initialization parameters are used for describing three types of interaction parameters. Since the number of parameters are uncertain, the parameters numbers of different models have great differences. It is necessary to set three types of parameters individually. Initialization parameters are configuration information when the model is called.

3.2 Combination Model Description

The combined model has basic properties information without expression, such as the model creator, publisher, version information, etc. However, the model information (input, output, initialization, etc.) occurs great changes. And the description of functions and related information also need to be redefined. Resource description form of combined model is shown in Fig. 3.

As showed in Fig. 3, combination model URI is a virtual URI. Based on the combination information, platform looks for its models and reproduces combination internal processes when the user calls combination model. The procedure parameters transitive relation is matched by the relation configuration between RelaitonFrom and RelationTo in combination model. Configuration information for all models is obtained from the corresponding initialization information.

3.3 The Incidence Relation Description

In the model ontology description, the model incidence rela-



Fig. 4 The model incidence relation description



Fig. 5 Resource description schematic diagram of simulation tasks flow

tion can't be shown directly because the incidence relation among models grows exponentially with the increasing of its number. Thus resource description file, which records the model incidence relation, requires separate treatment. Without considering the impact of input and output, the model incidence relation will comprise a two-dimensional relation if two arbitrary models exist incidence relation. Its resource description form is shown in Fig. 4.

As showed in Fig. 4, used times are the total frequency of calling the models. The remaining values are frequency of using two models together. The calculation method of model incidence relation degree is shown as follows:

$$d(M_A, M_B) = \frac{frequency of two models together}{total frequency of the models}$$
(13)

Compared with other MIRD, the most commonly used model combination is obtained by ranking. With some data mining methods [24], the MIRD can further improve the retrieval accuracy.

3.4 Simulation Tasks Flow Description

According to the definition of T_{test} , the description of simulation tasks flow includes simulation model, model transferring parameters information, the configuration parameters

of each model, the drive and property of flow, etc. The drive and property are configuration information, namely simulation task name, simulation step-length, simulation Start time and simulation End time. In order to reconstruct the flow, the information needs to express completely in simulation tasks flow. The basic description form is shown in Fig. 5.

As illustrated in Fig. 5, step is the simulation steplength. StartTime and EndTime are the starting time and ending time of the simulation tasks respectively. Compared with combination models, entry is the corresponding default value which was previously configured in the simulation task flow. And output item is the simulation results.

4. Architecture Design and Implementation of Model Management in GDSE

GDSE Model Management Center, one of the three terminals in simulation platform, plays a very important role as a result of a large number of information exchanges with the Human-Computer terminal or Service Simulation Model Terminal.

Model management center architecture of the simulation platform consists of four parts, namely the Semantic Retrieval Center, Model Registration terminal, Simulation Tasks Flow Records Center and Semantic Description File Storage Server (Fig. 6). Model registration terminal is primarily used for processing the registration information of simulation model and user. After registration in the platform, the metadata and registration information of the simulation model are processed in accordance with semantic rules (RDF, etc.), and stored in Data Recording Server. While designing simulation task, the user obtains model basic information through retrieving semantic search center. Based on the information, the user calls the models and designs simulation task. According to configuration task flow in human-computer interaction terminal, simulation tasks records center processes and updates the model incidence relation description file and simulation tasks recording file.

Based on the basic infrastructure of GDSE, we know that all models are registered in the platform after encapsulated. And they are provided with unified interface standards and calling conventions. The model description is mainly about simulation model, simulation task flow, etc. Description formed process and relations among different descriptions (Fig. 7). As shown in Fig. 7, all simulation models firstly need to go through the service-oriented encapsulation. External performance is service-oriented model ontology in the platform. During registering model in platform management center, ontology is mapped the service-oriented model metadata and added missing registration information. Metadata and registration information contain all the required described information of the service simulation model, including basic properties, related functional description, the input/output information, etc.

The model management center integrates model description information, user registration information, model parameter statistics, simulation task record information, in-



Fig. 6 The architecture of model management center in GDSE



Fig. 7 Description formed process and relations among different descriptions in simulation model



Fig. 8 The GDSE Human-Computer Terminal window

cidence relation and other information of the model. And it forms resource description system with semantic retrieval function and incidence relation, including four main descriptions, i.e. the model resource description, model incidence relation description, combined model description and simulation task flow description. The MIRD has a close relationship with the model resource. Simultaneously, the MIRD value is the statistical information of simulation task records. Model combination and simulation tasks flow are defined by its internal service-oriented model.

The model management module is implemented in GDSE as above architecture. The window of model retrieve on GDSE human-computer terminal is stated in Fig. 8. It can be seen that the model management can retrieve model through either requirement or keywords.

5. Simulation and Analysis

To analyse the performance of semantic management proposed in this paper, a Monte Carlo test was carried out with the comparison of traditional management method. The model is usually retrieved through keywords when using traditional management method. It will take a retrieval operation for one or two models. A simulation task may take 5-10 retrieval operations which needs 10-20 models. While a simulation task represent a requirement. The semantic management method proposed above can achieve the retrieval through requirement. It may take only one retrieval operation for all models needed by a simulation task. It will cost much less time than the traditional management method in theory. The statistical analysis of model retrieval through keywords and requirement is made respectively on the basis of model management module. For testing the retrieval ability, firstly, 5000 models information is registered. As showed in Fig. 8, secondly, the retrieval was performed respectively in two ways, viz. retrieval through requirement and keywords. Retrieval through requirement is the retrieval method of semantic management, while the retrieval through keywords is a retrieval way of traditional management. The retrieval time and results were recorded for analvsis. Section retrieval terms were listed in Table 1. Simulation task flows for testing usually consist of 8-20 models.

As showed in Fig. 9, retrieval time is recorded 200 times with different requirements in Table 1. The solid line is the retrieval time through requirement, and the dash-dotted line is retrieval time through keywords. It can be seen that the retrieval time is 50–150ms (maintained at around 100ms) through keywords. Retrieval through requirement needs a little more time (approximately 50 to 100ms) in one retrieval. However, it had little effect on the retrieval experience.

Figure 10 shows the percentage of the available model number (used in the task) in the model amount (needed by the corresponding task). The solid line represents the retrieval results through requirement, and the dash-dotted line represents retrieval results through keywords. It can be seen that the results through requirement could meet 70% needs of total simulation task, while that through keywords only meets about 10%–20%. It shows that the results of retrieval through requirement provides much more appropriate mod-

 Table 1
 A section retrieval terms through requirement and keywords

No.	Simulation Task	Requirement	Keywords
1	GPS single satellite	pseudrange	pseudorange\rinex
	pseudrang simulation	+ GPS	\time\
2	Static user	positioning	static\positioning
	positioning simulation	+ static	\coordinate\
3	INS\GPS Integrated	INS + GPS	INS\GPS
	navigation simulaiton	+ integrated	\filter\
4	Airplane user	positioning	airplane\filter
	positioning simulation	+airplane	\coordinate
	1 0	1	\positioning
5	Car user \Beidou	positioning	car\positioning
	positioning in GPS	+ car+GPS	\coordinate
	positioning in or o	+Beidou	\filter
		Deldou	(inter



Fig. 9 Cost time in a model retrieved through key words and requirement respectively



Fig.10 The percentage of retrieved model number in model amount needed by the task

els for simulation task in one retrieval. So the retrieval times will be greatly reduced.

Figure 11 shows the percentage of available results in one retrieval. Retrieval through keywords is about 10%-50% available model (Individual results may up to100%, which show that the results may all be available), and that through requirement is more than 50%. The retrieval results



Fig. 11 The percentage of available results in all retrieval results



Fig. 12 The retrieval times for all models needed

through requirement generally has a high utilization ratio than the results retrieval through keywords.

Figure 12 is the comparison of retrieval times by using two ways. Because the retrieval through requirement can't provide all models needed by users. The rest models still need to be retrieved through keywords. So in solid line, the first time is retrieved through requirement, and the rest times are retrieved through keywords. The dash-dotted line is the retrieval times only through keywords. It can be seen that the retrieval times through requirement are mainly about 1– 3 times (Individual 5 times), which are much less than that only through keywords.

In general, retrieval through requirement will get all models, which are needed by the simulation task, much more quickly than that through keywords. And more models needed by users, the more quickly than that through keywords. Based on the above test results, it is shown that the model management method has a certain ability to understand user needs and quickly get all models in the large-scale model library.

6. Conclusion

GDSE has a strong advantage in terms of sharing and reusing. Features of configurability and composability are

contained in a service-oriented simulation model. It has a complex incidence relation between models. The simulation task process, which formed during tasks design, has its uniqueness. Based on above-mentioned property and the study of RDF and OWL, resource description method and the basic architecture of model semantics management are proposed. Simulation results show that the method has a higher retrieval accuracy and efficiency than traditional management method for simulation task design. This method can effectively implement semantic management, achieve the understanding of user needs and the indexing of models, reduce the workload of the user retrieval, and improve the efficiency of the simulation tasks design. In order to further enhance the design efficiency of simulation task, majorization details of receipt semantic retrieval architecture, description method and reasoning abilities of model incidence relation, are further research.

Acknowledgments

This research was financially supported by the National Science Foundation, China (No.61502257).

References

- [1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communications of the ACM, vol.53, no.4, pp.50–58, 2010.
- [2] C. Baun, M. Kunze, J. Nimis, and S. Tai, Cloud Computing-Web-based Dynamic IT Services, Springer-Verlag, Berlin Heidelberg, 2011.
- [3] R. Buyya, R. Ranjan, and R.N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," High Performance Computing & Simulation, pp.1–11, 2009.
- [4] F. Dong and A. Malloy, "Recent research advances in cloud computing and big data," Concurrency and Computation Practice & Experience, vol.27, no.18, pp.5574–5576, 2015.
- [5] M. Ramachandran, "Software security requirements management as an emerging cloud computing service," International Journal of Information Management, vol.36, no.4, pp.580–590, 2016.
- [6] S.L. Wang, K.T. Liang, J.K. Liu, J.Y. Chen, J.P. Yu, and W.X. Xie, "Attribute-Based Data Sharing Scheme Revisited in Cloud Computing," IEEE Trans. Inf. Forensics Security, vol.11, no.8, pp.1661–1673, 2016.
- [7] Y. Cao, L. Yang, Y. Tan, H. Cao, and S. Cao, "Researches on an open and extensible healthcare information service mode based on service-oriented architecture," Basic & Clinical Pharmacology & Toxicology, vol.117, no.SI, pp.22–23, 2015.
- [8] F. Paraiso, P. Merle, and L. Seinturier, "soCloud: A serviceoriented component-based PaaS for managing portability, provisioning, elasticity, and high availability across multiple clouds," Computing, vol.98, no.5, pp.539–565, 2016.
- [9] P.Y. Reyes-Delgado, M. Mora, H.A. Duran-Limon, L.C. Rodriguez-Martinez, R.V. O'Connor, and R. Mendoza-Gonzalez, "The strengths and weaknesses of software architecture design in the RUP, MSF, MBASE and RUP-SOA methodologies: A conceptual review," Computer Standards & Interfaces, no.47, pp.24–41, 2016.
- [10] B.H. Li, X.D. Chai, and B.C. Hou, "Networked modeling & simulation platform based on concept of cloud computing-cloud simulation platform," Journal of System Simulation, vol.21, no.17, pp.5292–5299, 2009.

- [11] G.C. Fan, C.D. Xu, C.S. Hu, and D. Song, "Real-time intelligent recommendation method of a simulation model based on incidence relation," Chinese Journal of Engineering, vol.39, no.4, pp.626–633, 2017.
- [12] C.S. Hu, C.D. Xu, P.F. Zhang, and X.B. Cao, "Study on simulation model virtualization in cloud computing environment," Huazhong Keji Daxue Xuebao (Ziran Kexue Ban)/Journal of Huazhong University of Science and Technology (Natural Science Edition), vol.40, suppl.1, pp.135–140, 2012.
- [13] C.S. Hu, C.D. Xu, G.C. Fan, H. Li, and D. Song, "A Simulation Model Design Method for Cloud-Based Simulation Environment," Advances in mechanical engineering, vol.5, no.932684, 2013.
- [14] C.S. Hu, "Study on key techniques of integrated design environment for complex product based on Service-Oriented Architecture," Ph.D. thesis, Beijing Institute of Technology, Beijing, China, 2014.
- [15] A. Stellato, "Dictionary, Thesaurus or Ontology? Disentangling Our Choices in the Semantic Web Jungle," Journal of Integrative Agriculture, vol.11, no.5, pp.710–719, 2012.
- [16] F. Zhang, J. Cheng, and Z. Ma, "A survey on fuzzy ontologies for the Semantic Web," Knowledge Engineering Review, vol.31, no.3, pp.278–321, 2016.
- [17] J. Ashraf, O.K. Hussain, and F.K. Hussain, "Empirical analysis of domain ontology usage on the Web: eCommerce domain in focus," Concurrency & Computation Practice & Experience, vol.26, no.5, pp.1157–1184, 2014.
- [18] S. Powers, Practical rdf, O'Reilly Media, Sebastopol, CA, 2003.
- [19] N. Taghezout, N.S. Houari, and A. Nador, "Negotiation model for knowledge management system using computational collective intelligence and ontology-based reasoning: case study of SONATRACH AVAL," International Journal of Simulation and Process Modelling, vol.11, no.5, pp.403–427, 2016.
- [20] M.-Y. Choi, C.-J. Moon, and D.-K. Baik, "Transformation of a Relational Database to RDF/RDFS with ER2iDM," leice Transactions on Information & Systems, vol.96, no.7, pp.1478–1488, 2013.
- [21] R. Guns, "Tracing the origins of the semantic web," Journal of the American Society for Information Science & Technology, vol.64, no.10, pp.35–37, 2013.
- [22] J.G. Li, G.S. Zhao, C.M. Rong, and Y. Tang, "Semantic description of scholar-oriented social network cloud," Journal of Supercomputing, vol.65, no.1, pp.410–425, 2011.
- [23] J. Heflin, R. Volz, and J. Dale, "Requirements for a Web ontology language," W3C working draft, no.8, 2002.
- [24] J. Chen, L. Zhu, Y. Guo, S. Cai, and X. Zhao, "A mining approach for component abnormal information based on monitor log," International Journal of Simulation and Process Modelling, vol.11, no.5, pp.353–362, 2016.



Chunsheng Hu is a lecturer of the School of Mechanical Engineering at the Ningxia University (NXU), China. He received his M.S. and Ph.D. in Aeronautical and Astronautical Science and Technology from Beijing Institute of Technology (BIT). His research interests include: simulation of complex systems, cloud computing and satellite navigation.



Xueen Zheng is a Ph.D. candidate in School of Aerospace Engineering of Beijing Institute of Technology (BIT), China. His research interests are focused on simulation of complex systems and satellite navigation.



Chengdong Xu is a Professor of the School of Aerospace Engineering at the Beijing Institute of Technology (BIT), China. He received his B.S. degree in Mechanical Engineering from Zhejiang University and Ph.D. in Aircraft Design from Beihang University. His research interests include: complex system simulation, satellite navigation and aircraft design.





Guochao Fan is a researcher in Beijing Jinghang Computation and Communication Research Institute, China. He received his Ph.D. in Aeronautical and Astronautical Science and Technology from Beijing Institute of Technology (BIT), China, in 2018. His research interests are focused on simulation of complex systems and satellite navigation.