Industrial design based on computer aided simulation

Jianxun Wang*, Qingyun Zhou

Zhengzhou Technical College, Zhengzhou, 450121, China

Received 1 March 2014, www.cmnt.lv

Abstract

Recently, new technologies have emerged in industrial automation platforms. A rapid modelling and simulation environment is required to integrate these new technologies with existing devices and platforms to reduce the design effort and time to market. System-level modelling is a popular design technique that provides early simulation, verification, and architectural exploration. However, integration of real devices with system models is quite challenging due to synchronization and hard real-time constraints in industrial automation. Simulations are software tools approximating and predicting the behaviour of real industrial plants. Unlike real plants, the utilization of simulations cannot cause dam-ages and it saves time and costs during series of experiments. A shortcoming of current simulation models is the complicated runtime integration into legacy industrial systems and platforms, as well as ad-hoc design phase, introducing manual and error-prone work. This paper contributes to improve the efficiency of simulation model design and integration. It utilizes a semantic knowledge base, implemented by ontologies and their mappings. The integration uses the Automation Service Bus and the paper explains how to configure the runtime integration level semantically.

Keywords: assembly systems, estimation, part feeding, pose statistics, simulation

1 Introduction

Industrial automation systems are experiencing a paradigm shift due to incorporation of new technologies, such as embedded real-time devices and communication networks. It has been shown that the demand for plug and play mechatronic solutions increases significantly and the investments on automation systems that utilize hardware–software and communication infrastructures cannot be ignored in today's factory systems [1-3]. As the complexity of these systems increase, a rapid modelling and simulation environment is required to reduce the design and verification time. System-level modelling is a very effective way of reducing the development cycle while providing early prototypes and enabling architectural exploration and verification.

The role of simulation models is becoming more essential as industrial systems are being more complex and their structure more sophisticated. Simulation models are necessary for the advanced process control and they are important for optimization or as test-beds to analyze the behavior of both a control system and the whole plant under normal, extreme or testing conditions, when an engineer compares different parameters or control algorithms under various operation scenarios.

A lot of research effort has been invested into problems dealing with mathematical modelling, some effort has been in-vested into the description of models. But only few works have been concerned with the description of the whole automation system including knowledge about the real plant, control system, software tools, and communication interfaces not only on the hardware level, but also on the software level. Such pieces of information should be explicitly conceptualized, interfaces for the integration of sub-systems should be described in a machine-understandable form and the integrated solution has to be consistent and easy to maintain.

System-level modelling also allows reuse of different forms of intellectual properties (IPs), which is a common practice in the industry. Since industrial automation systems are increasingly connected with other IPs or industrial components, they should also benefit from system-level modelling techniques to reduce the development costs. In traditional system-level modelling, all components need to be modelled. However, modelling has no added value for components that are already physically implemented because modelling is an abstraction mechanism and requires human effort. Therefore, techniques have to be developed for incorporating real devices with virtual models. Traditionally, there are communication mechanisms between virtual models. These mechanisms are defined through the constructs of modelling languages. Similarly, real implemented devices communicate with each other through physical mediums with predefined exchange data formats. However, there is no established mechanism that provides communication between real devices and virtual models, and this paper fills this gap. Specifically, we present synchronization mechanisms between virtual models and real devices so as to achieve real-time communication.

The goal of this paper is to introduce an ontology-based approach for the plant description that is focused on both

* ***Corresponding author's*** e-mail: wangjianxuanzz@126.com

complex and flexible automation systems. Although this paper deals especially with simulation models, the approach provides additional support for integration of diverse engineering tools that are involved in the automation system engineering as well. All phases of the automation system life-cycle are based on the description of the real industrial plant. In the early design stage, such a description is needed for optimization of the process structure and parameters, afterwards, it is useful to design a simulation model as a virtual test-bed for the control system design and fine-tuning. Since every industrial process is modified and maintained, the plant description has to reflect the current state. Particularly, modifications of the industrial process bring fundamental benefits of the explicit plant description that is independent on other engineering tools, such as simulators. For example, in approaches used nowadays a simulation expert has to modify the simulation model according to the modified plant, which is usually time-consuming, because he or she has to study principles of the simulation model, meaning of required parameters and often empirically check the results. Using the explicit plant description a process specialist modifies this description according to the real plant and all necessary changes are propagated into the simulation model and other systems in the automated way. Traceability of changes is one of the important features of modern system engineering.

This paper addresses problems related to the integration of automation systems including simulations, human-machine interface (HMI) as a part of SCADA systems (standing for supervisory control and data acquisition), and interaction with real systems solved via OPC UA or OPC classic. Some tools can be legacy, i.e., they had been designed before the need for interoperability emerged. One of the automation system tools are simulations, being the software approximations of the real plant behavior. Since their design phase, reuse and access to historical data pose problems which have not been satisfactorily solved by now, this paper addresses these issues. The presented simulation model design is oriented on structural issues dealing with simulation blocks, their interconnections, and parameters; whereas mathematical equations are not handled.

The proposed approach utilizes ontologies comprising knowledge about design and integration of a real plant, a simulation model, as well as other industrial automation tools. The knowledge base provides the information in a machine-understandable form, hence the computer is able to process and query the knowledge required for the integration and the simulation model design. The presented approach is an application of the semantic integration [4-7]. In contrast to classic integration approaches, the semantic integration is based on utilizing mappings between adequate entities. For example, semantic map-pings relate really measured variables and their simulated approximations, real devices and their equivalents in a simulation model or local tag names used in a particular tool with the global representation of tag names.

2 Related Works

In the last decade, there have been significant studies to develop design platforms for intelligent industrial automation systems [8-10]. These automation systems rely heavily on a distributed computer-based infrastructure, where smart sensors and actuators, intelligent machines, robots, and other automation devices can interact using industrial protocols and take decisions in real time. In these systems, system-level communication, device synchronization, and the integration of new devices to the system are extremely challenging issues. Hence, there is a demand for sophisticated tools for the design of intelligent complex industrial automation systems. In SIMOO-RT [11-13], an object-oriented framework is pro-posed for modelling a real-time industrial automation system. A system model with real devices cannot be developed with SIMOO-RT. OOONEIDA [14-16] complies with IEC 61449 [17], which is a standard to design distributed control systems with intelligent devices. It proposes encapsulation of different types of IPs into reusable portable software modules called as function blocks. To achieve this, interfaces are created between various kinds of automation IPs such as devices, RTOSes, machines, systems, and industrial enterprises. However, in IEC 61499, the real-time properties of applications (e.g., reaction time) and resources (e.g., polling of data by function blocks, communication) are unspecified [18]. In OOONEIDA, real-time properties have to be handled via embedded controllers that are introduced to the system by encapsulation. There has also been a proposal on handling real-time issues in function blocks [19], but it has not been implemented yet. The encapsulation modules are developed in Java for the Eclipse and NetBeans integrated development environments. In RIMACS [20] project, a service-oriented architecture with real-time capabilities is proposed. Temporal behavior of each activity is isolated as much as possible by using dedicated hardware and software [4]. Linux kernel is modified to provide temporal isolation.

On the technical interoperability level, there can be found approaches using general-purpose distribution techniques such as DCOM, CORBA, J233, as well as architectures or frameworks related to the simulation area, such as DIS, SEDRIS, HLA. Especially HLA is used in many approaches, but its shortcomings are the lack of a semantic layer and data source description. The ex-tension of HLA with semantics is proposed in [12]. Ref. [114] proposes the Ontology Driven Simulation Design Tool (ODS). It is based on two ontologies that are mapped: the domain ontology categorizing knowledge including a problem vocabulary, and the modelling ontology used for the simulation model description.

3 Motivation and Formulation

The major challenges for the automation system integration in runtime are expressed in Figure 1. It depicts the typical architecture used in the industrial practice now.

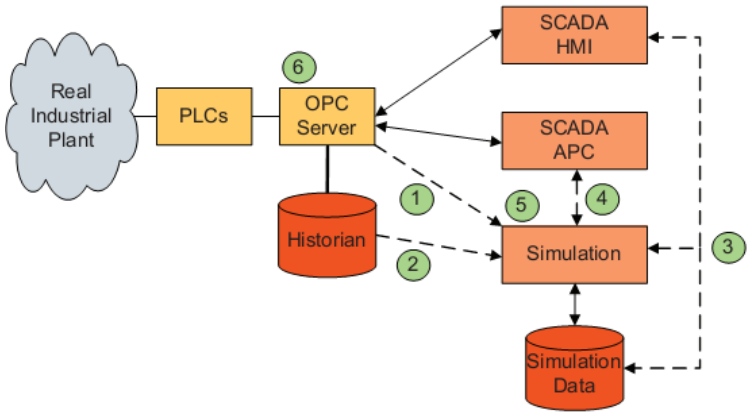

FIGURE 1 Current approach in industrial automation practice and its challenges

The meaning of circled labels refers to the issues as follows:

1. Import of runtime data into simulations.
2. Import of historical data into simulations.
3. Visualization of simulation data in standard HMIs and testing dispatchers' commands entered via HMI on simulation models.
4. Testing of SCADA advanced process control (APC) control actions on simulations.
5. Semi-automated design of simulation models.
6. Semi-automated configuration of OPC tags and other integration interfaces.

Analyzing these major challenges, our approach is based on two integration levels:

1) The technical integration level dealing with the technical transmission of data between particular tools.
2) The semantic integration level, which captures the meaning of the relevant knowledge, mapping common pieces of knowledge, and configuring the technical integration level. The paper addresses especially the semantic level of integration, whose research issues are summarized in the following overview.

The problems, which are investigated in this paper, were summarized into following research issues. Simulation models are hard to design and integrate manually. The goal of this research issue is to design an approach, which is based on explicitly represented engineering knowledge and to support the design of simulations semi-automatically. Since the runtime integration requires the proper structure of data in design time, the knowledge base must respect the integration request and involve it into the design time as well.

Current technical integration approaches require repeating manual work. This research issue is aimed at supporting runtime integration of industrial automation tools with a strong focus on simulations, which is flexible and easy to configure. Since the Simulation Framework is used as the technical integration level in the presented approach, one of the required output of the knowledge base tool is a generated set of configuration files for the Simulation Framework.

4 The Design of simulation based model

Although simulations are becoming widely used in industrial practice, their integration within the remainder of the automation system has not been satisfactorily solved. In addition, the design phase of simulations is usually done ad-hoc, which causes problems when either the real system is modified or some parts of the simulation should be reused. The goal of the presented work is to improve the design phase of simulation models in order to integrate them within the remainder of the automation system easily. We propose to capture the knowledge required for the simulation model design in the knowledge base.

4.1 SEMANTIC DESIGN OF SIMULATION MODELS

The typical engineering workflow [16] starts with the under-standing and summarization of a terminology. In other words, a domain specific language (DSL) is adopted for a particular class of plants, such as for hydraulic systems. DSL includes typical components and possible kinds of their connections, typical parameters and their types, relevant physical variables, definitions of relevant mathematical models, etc. Such pieces of information have to be structured and stored in a real plant ontology.

Plant device classes are represented as ontology classes, their parameters and relationships as ontology proper-ties. The creation of a general plant ontology structure enables to accelerate further project stages and it is done only once for every type of plant.

Having the ontology structure for the real plant, the description of the particular physical plant is created, i.e., there are created individuals of ontology classes. The individuals are labeled with unique names and relations to other individuals are entered. Parameters and their values can be entered, too. Based on the real plant ontology, simulation specialists are able to implement a simulation library, comprising generic simulation blocks. In a basic case, every plant ontology concept corresponds to a simulation block; in a more general case, a real plant element can be modeled by several blocks or vice versa. In the further subsection, this problem is discussed in more details.

The methodology for simulation model design and integration strongly depends on the available knowledge at the initial state of the automation project. In simulation practice, it is usual to utilize libraries containing generic blocks that are parameterized to approximate a specific device. In most cases, engineers are not able to create the library from scratch and afterwards assemble the simulation using these blocks. Usually, they must gain experiences on early versions of the simulations, perform a set of tests and comparisons of measured and simulated data and finally create or finalize the simulation library. Therefore, the first project in a specific area, such as a water distribution network simulation, requires iterative work on the simulation model and on the library. Other simulation projects of such networks reuse the library and even if some blocks must be added or modified, it is a simple task for an experienced simulation expert.

The simulation design phase is strongly influenced by the availability of the simulation library. If the simulation library exists, the semantic knowledge base tool is able to generate the simulation model semi-automatically. If the library is not available at the beginning, the knowledge base tool must be used in a similar way as an expert system, driving the user throughout a workflow collecting relevant pieces of knowledge required for the simulation model design.

In both cases, the general knowledge about the type of a real plant must be captured in the automation ontology. For example, the general knowledge about water distribution networks contains information that plants can have pumps, pipes, tanks, water wells, consumers or some disturbances; pumps have flow and pressure on their input and flow and pressure on their output, real parameters can be length, diameter and elevation, etc. The general knowledge is a kind of knowledge skeleton, which can be filled with real values when describing the specific plant. If needed, it can be extended with other parameters which are device-specific.

If the simulation library is available, the library blocks are annotated in the knowledge base, including their inputs, outputs and parameters. Consequently, the plant description must be formalized, including real plant topology and parameters. Technically, this step means populating the automation ontology with individuals denoting the real devices, their interconnections, tags, etc. Based on those pieces of knowledge, the semantic knowledge base tool is able to generate the simulation model structure automatically. Finally, the simulation expert is required to insert values of simulation block parameters, such as diameters or lengths of pipes.

In the case that the simulation library does not exist, the general knowledge is extended with a particular plant description in a similar way, but afterwards, the simulation model structure cannot be obtained automatically. The process of gathering the expert knowledge starts up with a specification which devices will be simulated. The step includes both grouping devices into complex subsystems (such as the whole plant can be decomposed to several simulation modules or a group of devices can be modeled as one block) and specification of block interfaces (such as the utilization of just one of the variables flow and pressure or both of them, an extension with further signals such as control values for devices). Afterwards, for each simulation block, there must be declared simulation parameters.

4.2 BLOCK-BASED DESIGN OF SIMULATION MODELS

System analysis and synthesis is typically based on the decomposition of problems into sub-problems. Engineering disciplines usually utilize components or blocks for the system design. For example, software engineers use methods as well as classes to encapsulate specific functionalities. In the simulation design area, the behavior of devices is often encapsulated by simulation blocks, which represent for example pumps, pipes, etc. Such blocks are comprised in a (universal) simulation library and the implementation of these blocks is generic in terms of their parameterization for the specific use.

This section addresses the research issue RI-2, i.e., the integration of whole simulations and other tools into the automation system environment. On the semantic level, the proposed solution is based on the representation of knowledge about local tag data models and mapping them to the global tag representation in the ontology. On the technical level, the approach is based on the enterprise service bus (ESB) approach. Although this paper is focused on the semantic level, the following paragraphs provide an overview of the technical layer, which is crucial for under-standing.

The proposed integration architecture, implemented by the Simulation Framework, is depicted in Figure 2. Tools are interconnected via connectors (having domain specific and tool specific parts) to the ASB. The ASB transfers data according to the defined service bus workflow. This architecture poses a solution to the technical integration, nevertheless, it must be configured via a set of XML files to work properly. The preparation of such configuration files can be solved either manually, bringing short-comings in repeating error-prone manual work, or semi-automatically. We propose the semi-automated approach, which guarantees a consistent solution being flexible and capable to adopt modifications of real plant or software automation tools easily. The proposed solution is based on the structural tag description based on mappings between global tag representations and their local tool names.

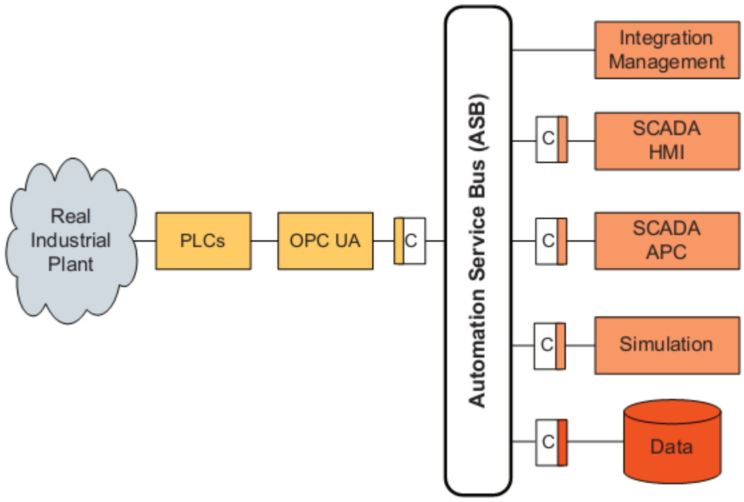

FIGURE 2 Industrial integration based on automation service

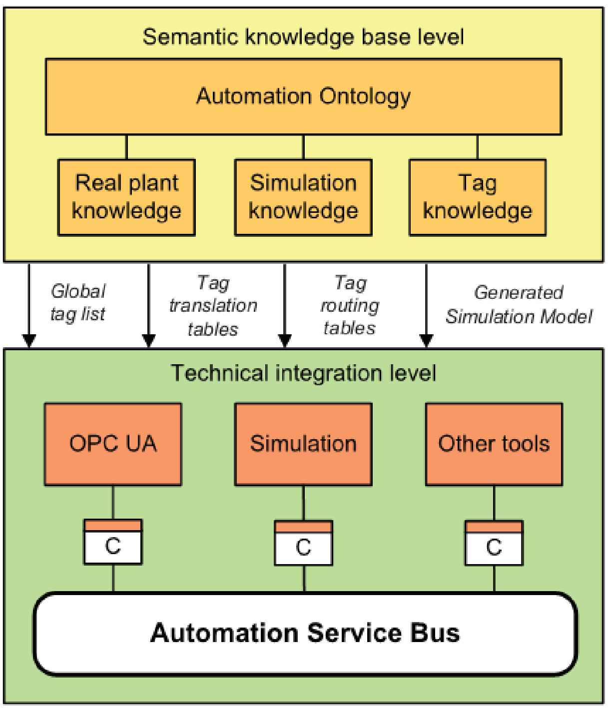

FIGURE 3 Technical layers including their interface

On the top of the Simulation Framework, we utilize the semantic knowledge base tool, which encapsulates the included ontologies, and provides interfaces to access it easily. The conceptual diagram of the proposed layered architecture is depicted in Figure 3. The knowledge base provides configuration XML files for the Simulation Framework, which are responsible for the definition of runtime behavior and furthermore, the knowledge base is able to support the design of particular tool models (such as simulation models in the current version).

4.3 PARAMETER MANAGEMENT OF THE SIMULATION SYSTEM

Since simulation parameters, i.e., parameters required for the parameterization of simulation blocks, can differ from parameters of real plant devices in both count and scale, the descriptive paradigm has to reflect this fact and manage them efficiently.

The presented approach distinguishes 3 classes of parameters. Real system parameters are parameters of devices included in the real system. Simulation model parameters are parameters of simulation modules. Simulation solver parameters are parameters of the simulation solver, such as simulation start time or stop time, required precision, etc.

In order to guarantee the consistency and avoiding errors while the real plant is changed, the presented approach involves the parameters in ontologies: the simulation ontology comprises definitions of required parameters to run the simulation for each block and plant ontology involves especially real plant elements description. A process specialist, for example, is not interested in the model parameters of a pump, which is modeled and simulated as the first-order system. Although he knows almost everything about the real pump, he does neither know how simulation specialists would model pumps, nor the expected value, that can be calculated even by some algorithm. On the other hand, to reach the consistency in large-scale projects, changes should be automatically propagated between engineering domains and tools, hence these parameters are inter-linked. Mapping corresponding entities, such as real system parameters and simulation model parameters, which depend on them, is very easy to handle in ontologies that is one of the advantages of using them in this area.

The configuration of the Simulation Framework is based on the following XML files: Global tag list, tag translation tables, and tag routing tables. The global tag list file is aimed at setting all existing tags in the ASB environment and their properties (such as type, minimal and maximal values or others, which are useful for a control system). The tag translation tables are related to ASB connectors, which translate tag names between local names (occurring in the particular tool data model scope) and global tag representations, being uniquely available in the ASB. The tag routing tables define how tag values should be distributed between the tools (such as simulation results must be transmitted to HMI, simulation inputs must be entered with an operator-training data-set). Note that the global tag list and tag translation tables are project-specific, whereas routing tables are task-specific. Every project can have several tasks, such as simulation can be used as a soft-sensor to estimate unmeasured states, or as a test-bed to analyse control system behavior and train operators.

5 The real simulation cases

We selected a sample project to test the presented approach and to evaluate its efficiency. Since the complexity of typical industrial systems is very high and too complicated for a conference paper, we selected the simulation of the passive house. The goal is to describe a particular passive house in the unified ontology way and to generate its simulation model automatically. The simulation library with generic blocks is available. It is called Bldsimlib library [22], and it is implemented in MATLAB-Simulink. To create a simulation model of a particular house, the appropriate generic blocks from the library bldsimlib.mdl have to be inserted into a blank simulation model file simulation.mdl. Their names have to be set uniquely, and the blocks have to be interconnected according to the passive house floor plan. As well, the required parameters of the blocks have to be entered into the simulator workspace, in our case MATLAB Workspace.

For the first experiments, a simplified passive house was defined. It offers enough testing possibilities and the obtained results can be easily checked by people. To describe the testing house, a passive house ontology was designed. Its concepts define general elements of passive houses, such as rooms, walls, windows, etc. Afterwards, a description of the testing house was created by making ontology individuals representing the real passive house equipment.

The task for engineers is to design a simulation model for this plant and to integrate it within the remainder of the automation system. According to the workflow presented in Figure 2, the simulation model design requires general knowledge about hydraulic systems at the beginning. The second step is to populate the knowledge base (i.e. the automation ontology) with individuals representing devices in this particular industrial plant. In case of the universal simulation library being available, the simulation model structure can be generated automatically; otherwise, the simulation expert is driven through the process of gathering relevant simulation data. In this case, the simulation expert selects which devices create a simulation module and which blocks will be included in the modules. Since the system does not include repeating segments such as pump-stations, every device will be approximated by one simulation block. Interfaces of the blocks will be input and output flow and pressure; furthermore, tanks will have level sensor variables as outputs. Pumps and valves will have input control signals to distinguish their states. As tags are a subset of variables, there are expected tank level tags and pump flows to be considered as tags. The subsequent step in the methodology is to define simulation parameters. One possible solution is to accord to every device an elevation to the reference point and to define for every tank its height, diameter, and diameters of inlets and outlets. Pipes are expected to have resistances and pumps maximal volumetric flows. In a further step of the design methodology, the simulation library will be derived automatically. Finally, the simulation parameter values must be entered.

6 Conclusion

Industrial plants and processes are becoming very complex and sophisticated. Appropriate control algorithms, being necessary for their efficient operation, can be tested and fine-tuned on simulation models; for advanced process control the models are even necessary. Therefore, the fundamental precondition for control algorithm design is to implement and fine-tune a simulation model. Since plant description methods used nowadays are insufficient for describing modern flexible plants, this paper presents an ontology-based approach to the plant description. The proposed paradigm enables querying and inferring new pieces of knowledge. The solution supports the efficient integration of diverse engineering tools, such as simulators, SCADA systems, or even proprietary software used in particular projects. The proposed approach supports reasoning techniques and efficient utilization of domain specific languages for each engineer or expert. Ontologies comprise explicitly specified knowledge, enable to interchange it between automation tools and can guarantee consistency of the whole integrated solution. The presented approach can be used for both continuous and discrete systems, especially the large-scale ones. A practical example, describing a passive house with ontology individuals and generating a structure of a simulation model automatically by the implemented tool, is involved in this paper.

References

[1] Moser T, Mordinyi R, Sunindyo W, Biffl S 2010 Semantic service matchmaking in the atm domain considering infrastructure capability constraints *Canadian Semantic Web* 133-57

[2] Gruber T 1993 A translation approach to portable ontology specifications *Knowledge Acquisition* **5**(2) 199-220

[3] Gómez-Pérez A, Fernandez-Lopez M, Corcho O 2004 Ontological Engineering with examples from the areas of Knowledge Management e-*Commerce and the Semantic Web second printing ed London* Springer-Verlag

[4] Obitko M 2007 Translations between ontologies in multi-agent systems *PhD dissertation Czech Technical University Faculty of Electrical Engineering* Prague

[5] Lacy L, Gerber W 2004 Potential modelling and simulation applications of the web ontology language – OWL *Proceedings of the 2004 Winter Simulation Conference* **1**

[6] Silver G, Hassan O H, Miller J 2007 From domain ontologies to modelling ontologies to executable simulation models *Proceedings of the 2007 Winter Simulation Conference* 1108–17

[7] Hu J, Zhang H 2009 Ontology based collaborative simulation frame-work using HLA and Web Services *World Congress on Computer Science and Information Engineering* **5** 702-6

[8] High Level Architecture (HLA) 2003 US Defence Modelling And Simulation Office

[9] Sarjoughian H, Huang D 2005 A multi-formalism modelling compos-ability framework: agent and discrete-event models *Proceedings of Ninth IEEE International Symposium on Distri-buted Simulation and Real-Time Applications* 249-56

[10] Benjamin P, Akella K Towards ontology-driven interoperability for simulation-based applications *Proceedings of the 2009 Winter Simulation Conference* 1375-86

[11] Durak U, Güler S, Guztüzün H, Ider S K 2007 An exercise in ontology driven trajectory simulation with MATLAB SIMULINK (R) *Proceedings of the 21th European Conference on Modelling and Simulation* Prague

[12] Barth M, Fay A 2013 Automated generation of simulation models for control code tests *Control Engineering Practice* **21**(2) 218-30

[13] Silver G, Bellipady K, Miller J, Kochut K, York W 2009 Supporting interoperability using the discrete-event modelling ontology (DeMO) *Proceedings of the 2009 Winter Simulation Conference* 1399-410

[14] Silver G, Lacy L, Miller J 2006 Ontology based representations of simulation models following the process interaction world view *Proceedings of the 2006 Winter Simulation Conference* 1168-76

[15] Vrba P, Tichý P, Mařík V, Hall K, Staron R, Maturana F, Kadera P 2011 Rockwell Automation's holonic and multiagent control systems compendium *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews* **41**(1) 14-30

[16] Novak P, Sindel R 2012 Semantic Design and Integration of Simulation Models in the Industrial Automation Area *2012 IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA)* 1-8

[17] Melik-Merkumians M, Zoitl A, Moser T 2010 Ontology-based fault diagnosis for industrial control applications *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA)* 1-4

[18] Novak P, Sindelar R2011 Applications of ontologies for assembling simulation models of industrial systems *On the Move to Meaningful Internet Systems: OTM 2011 Workshops* Hersonissos Springer Dordrecht 148-57.

[19] Gawthrop P, Bevan G 2007 Bond-graph modelling *IEEE Control Systems Magazine* **27**(2) 24-45

[20] Sindelar R, Novak P 2012 Simulation integration framework *The 10th IEEE International Conference on Industrial Informatics (INDIN 2012)* 80-5

Authors

Jianxun Wang, born in May 1972, Zhengzhou, Henan Province, China.

Current position, grades: associate professor at Zhengzhou Technical College, China.
University studies: Master's degree in Mechanical Design and Simulation.
Scientific interests: mechanical design and simulation.
Publications: 15 papers.

Qingyun Zhou, born in September 1972, Zhengzhou, Henan Province, China.

Current position: associate professor at Zhengzhou Technical College, China.
University studies: Master's degree in computer science and technology.
Scientific interests: computer software, system simulation and automatic control.
Publications: 12 papers.