Towards a User-Centric Service Integration Approach

Pengwei Tian, Yaoxue Zhang, Yuezhi Zhou, Ming Zhong, and Cunhao Fang

Key Laboratory of Pervasive Computing, Ministry of Education
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, China

Service Integration plays an important role in Service-Oriented Architecture (SOA). With the thriving of SOA, many approaches have been proposed to address the issue. Several languages or specifications are introduced for the definition of integration plan, such as WSFL, BPML and BPEL4WS. And many recent researches employ semantic web techniques to enable the automatic service discovery and interoperation, OWL-S included. Most of the existing approaches are originally introduced for application developers or programmers to use, in which user requirement is usually expressed based on the services themselves. However, a service may be very complex, the functions of which are usually the composite and interconnection of several basic functions. It is sometimes difficult for normal users without developing skills to understand the services and then utilize them to define the integration solution. In the paper, we think that, the most familiar things for normal users are the functions they need and the working process they expect. Based on the intuitive idea, we propose a normal User-Centric Service Integration approach (UCSI). In the approach, a set of function elements are predefined for each specified service domain and three working process notations are introduced to model the relationships between function elements. Then user requirement and service function can both be represented as function elements interconnection based on the process notations. With the representation, the functionality relationship between user requirement and services can be resolved and the service integration solutions can be generated. Normal users with certain knowledge of specified service domains can try using the approach to achieve personalized service integration. And a prototype system is introduced in the end.

Formal Model-Driven Engineering of Distributed Simulation Systems based on Architecture-Centric Domain-Specific Approach

Di Wu*, Jie Chen†, Flavio Oquendo‡

*School of Information Science and Technology, †Beijing Institute of Technology, Beijing, China
‡Computer Science Laboratory Valoria Université de Bretagne-Sud

Di.Wu@univ-ubs.fr, Chenjie@bit.edu.cn, Flavio.Oquendo@univ-ubs.fr

The Distributed Simulation (DS) systems possess the characteristics of functional and geographical distribution for participants; highly dynamic performance, huge amounts of runtime interaction, and openness for joining and resigning the system execution.

The software development for these systems is considered to be difficult in the aspects of domain-specific knowledge learning, architecture specification, architecture analysis, architecture refinement and final execution code generation.

This poster presents a new domain-specific approach for DS system development. It complies with the Model-Driven Engineering (MDE) paradigm and makes the software architecture as the central object during the development process. According to the requirements of DS systems, a new domain-specific language has been designed to cover the procedure. This language is based on a serious of formal architecture description languages, π-ADL, π-AAL and π-ARL.

An example of this approach will be showed in the poster for an IEEE DS system standard, the High Level Architecture (HLA). It aims to seamlessly cover the whole process of HLA system development centered by their architectures. This example includes architectural style definition, architecture specification, analysis, refinement to the final execution code. The corresponding toolset have also been developed to support for the easy use.

This method gives a complete development process for DS systems based on a serious of formal languages which provide a strong mathematical basement for system analysis and verification compared to other semi-formal languages. And at the same time, its ability of describing architecture internal behaviors gives more information of the system dynamic performance on the architecture level.