Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DESIGN AND IMPLEMENTATION OF CO-SIMULATION PLATFORM BASED ON .NET

[1,2]BIN PEI, [1]HUI HAN, [2]XIAOJIANG HOU, [2]YING SHENG, [2]BINGYI QIU

[1] The State Key Laboratory of Complex Electromagnetic Environmental Infects on Electronics & Information System, China

[2] 63888 Troop, Jiyuan, China

ABSTRACT

This paper analyzes the key technology of research and development on distributed co-simulation platform based on .NET, and puts forward architecture of distributed co-simulation platform based on .NET. The design aim of the frame is: fit for different platforms, conform to HLA standards and support distributed collaborative simulation, design a generality simulation and modeling platform of stronger extend ability so that the existing application can be easily integrated and deployed. The paper also introduces the method which uses XML and Web servers to realize the high distributed application in the internet, and explains some key technologies realization methods including using current simulation model with DLL and COM, simulation running control based on HLA, realization the collaborative communication with XML, and run manage of collaborative simulation.

Keywords: *.NET, Web Service, Simulation Model, Collaborative Communication*

1. INTRODUCTION

Co-simulation is used to make experiments and get information on the behavior of heterogeneous systems aiming at the validation of their design or at the evaluation of their performance. The large varieties of technologies make it difficult to conceive a generic mechanism, which promotes the cooperation between heterogeneous simulators. The complexity of current application systems have raised to a high level, these systems usually aggregate hardware and software cooperating parts. Their designs ask for models combining descriptions at different abstraction levels, and multiple specification languages are needed for representing those parts and/or abstraction levels [1]. Moreover, these parts are usually developed and validated by means of separate design processes. At various steps of these processes, however, it is necessary to verify if these parts interact correctly, in order to validate the overall functionality of the system. The validation of the whole system must take communication and cooperation aspects into account and may be extremely complex because of the system heterogeneity. In co-simulation environment, each simulator may send and receive data through a co-simulation interface that must handle communication, synchronization, and data format conversions. .NET and its relevance technologies provides a better platform for co-simulation of heterogeneous systems, this paper introduces the key technologies of co-simulation platform and also discusses those technology's realization methods.

2. AN OVERVIEW OF .NET

.NET is developed based on distributed computing, so it used some technologies of distributed computing, such as XML SOAP and so on. The main characteristic of .NET is:

(1) Applications are hardware-independent

All these languages are compiled via an intermediate binary code, which is independent of hardware and operating systems. This language is MSIL: Microsoft Intermediate Language.

(2) Applications are portable

Applications compiled as intermediate code are presented as Portable Executable (PE). .NET platform can be full or partial implemented over a vast range of hardware and software architectures.

(3) All languages must comply with a common agreement

To add a language to .NET, what is required in theory is for it to meet the requirements of the CLS (Common Language Specification), and for someone to develop a compiler from this language into MSIL. The fact that all the .NET languages are compiled in the form of an intermediate code also means that a class written in a language may be derived in another language, and it is possible to instantiate in one language an object of a class written in another language.

Web services is one core technology of .NET, the goal of web services is to enable a software application to embed references to the available libraries of functions, data, and other software application on other computers in internet. Web services enable interoperability by developers to produce functions for reuse and to consume functions created by others anywhere on a network.

In .NET, Web service interfaces are usually specified using the Web Services Description Language (WSDL). Defined using XML, WSDL provides a standardized way to specify the operations in an interface, the input and output parameters of those operations.

Once an interface has been defined, some protocol must be used to invoke the operations in that interface and return any results. The most common protocol for invoking Web services is the Simple Object Access Protocol (SOAP). SOAP, which is defined using XML, also relies on XML to define a standard format for the data protocol conveys. UDDI (Universal Description, Discovery, and Integration) allows developers of Web service clients to locate the information they need, such as WSDL interfaces, to build clients that can use appropriate Web services.

Because of the variety of implementation technologies, previous attempts at distributed computing (CORBA, DCOM, RMI) have yielded systems where the coupling between various components in a system is too tight to be effective. Standardized, XML-based web services enable a much more powerful and effective programming solution within traditional programming languages

3. CO-SIMULATION PLATFORM AND ITS KEY TECHNOLOGIES

3.1 Co-Simulation Platform

Co-simulation platform is a simulation integrated framework based on standard, which can support co-simulation between distributed, heterogeneous simulation models, and has the character of open and extended. Co-simulation platform is based on the object-oriented modeling and simulation method of continuous and discrete event mixed system, it provides the support of modeling methods and software environment for system modeling and integrating various modeling tools[4].

3.2 Key Techniques of Co-Simulation Platform

3.2.1 Modeling of co-simulation

In the co-simulation environment, various special people may master his familiar method of the system description; the special modeling method can better represent the various aspect of the system. Moreover, Aim at different simulation problem, the abstraction levels of the model's components are difference, using one model description can't satisfied with the demand of simulation. In order to make the conveniently cooperate between technologist and accurate description system, co-simulation environment must based on unified object model construction and support modeling method on mixed heterogeneous levels[4].

3.2.2 The communication of cooperation information

The communication of distributed network is challenged by the problems, such as heterogeneous physical environment, limited bandwidth of network, real time request. Moreover, the co-simulation environment involves a great deal of data, include model, simulation result and so on, and those datum have his own character, it is the key factor to use one data communication method, which can make effective communication, it should base on standard, easy to be realized and flexible extended data format.

3.2.3 Running management

In order to cooperate the simulation running between simulators, it is necessary to design a running manager to monitor the work state of the simulators. The manager also manages the simulation time and cooperate the simulation process, record the datum of simulation process. It can review simulation process when simulation is over [4].

4. OVERALL ARCHITECTURE

4.1 Design Goal

The goal of co-simulation platform is: fit for different platforms, conforms to HLA standards and support distributed collaborative simulation, design a generality simulation and modeling platform of stronger extend ability and which can be easy deployed.

4.2 Overall Architecture

In order to realize the goal of design, this paper presents the architecture of co-simulation platform as figure 1. The developers design the simulation model by co-simulation platform and other commercial software, such as Matlab, System view, and deploy those models to the models libraries which store at Web Service. Consumers can use co-simulation platform to search for the demand models from Web Service, and perform the simulation work on the platform. During the simulation time, the platform use HLA/RTI to control the process of simulation running, and monitor the simulator's state by running manager. Collaborative communication of all simulators can deal with the recognition, transform and synchronization of simulation datum. The platform can perform the evaluation task after simulation.

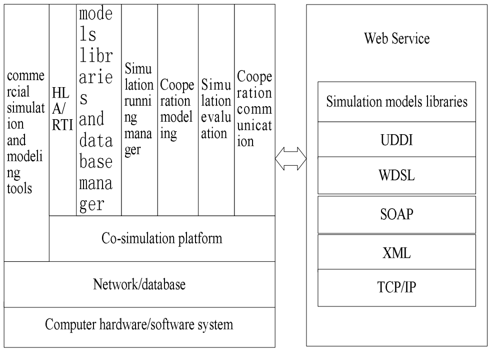

commercial simulation and modeling tools
HLA/RTI
models libraries and database manager
Simulation running manager
Cooperation modeling
Simulation evaluation
Cooperation communication
Co-simulation platform
Network/database
Computer hardware/software system
Web Service
Simulation models libraries
UDDI
WDSL
SOAP
XML
TCP/IP

Figure 1: Architecture Of Co-Simulation Platform

5. REALIZATION TECHNOLOGY OF CO-SIMULATION PLATFORM

5.1 Simulation Modeling

In order to meet the demand of model recognition and using, a generic specification of model configuration should be defined, it prescribes the naming and realization rules of interior interface function, the main functions is:

Model Initialize, defines basic function characteristics, including sample times, initial conditions of continuous and discrete states, and the sizes array.

Model Derivatives, calculates the derivatives of the continuous state variables.

Model Updates, discrete states, sample times, and major time step requirements.

Model Outputs, Calculates the outputs of the function.

Model Terminate, Performs any necessary end-of-simulation tasks.

The co-simulation platform perform modeling task by the style of flow chart, the system simulation model can be easy built by mouse drag operation.

5.2 Simulation Running Control Based on HLA

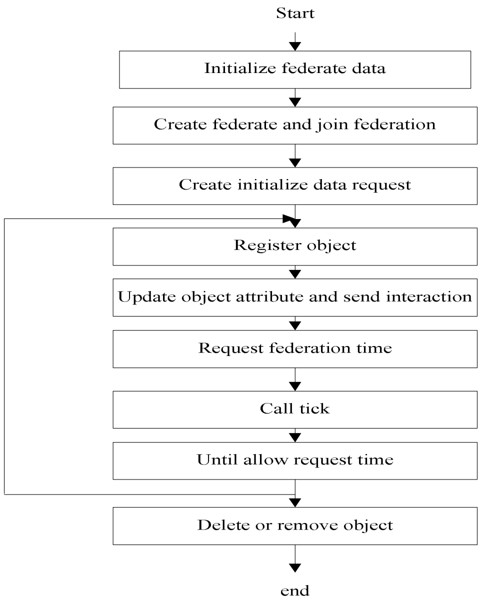


Figure 2: HLA Member Running Time Framework

HLA/RTI offer an environment of simulation software, it can connect the object which accord with the simulation request [3]. RTI is the core of co-simulation platform, it can provide the function such as: user login, managing varieties sub modules of simulation, the operation of object class (like creating new object, updating object, edit the attribute of object, deleting the object), managing of simulation time. The HLA member running time framework is as figure 2.

5.3 Collaborative Communication Based XML

The format of communication datum which transfer between Simulators in Distributed system can conform to the standard definition of PDU in IEEE1278.1-1995.The Structure of PDU is very complex, In order to support convenient access of upper levels application, object-oriented class encapsulating technology can be used to encapsulate the data structure and operation into of each PDU into a class, that can hide details of lower level's communication to upper levels, which is convenient for the development of upper levels. Owing to the encapsulating of PDU, the interface can keep fixed when the structure of PDU changes, it is beneficial to expand and modify the simulation system.

XML (Extensible Markup Language) is the collaborative work standard between Web Services and co-simulation platform, it is applied to data share and communication. It is easy to realize the data transform between XML and PDU object by the method of serializing and deserializing.

5.4 Compatible to Current Simulation Models

There are a large number of available models in current simulation system; those models are developed by business or user and stored in the style of DLL function libraries, COM component, or other type of file, the other type of file can be built into DLL or COM by the tools provided by commercial simulation software. DLL and COM can be integrated into Web Service; users of heterogeneous system can reuse those models. The method of Compatible to current simulation models is as figure 3.

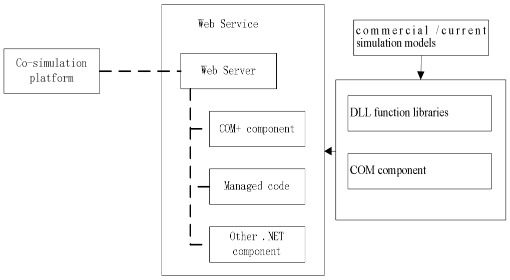


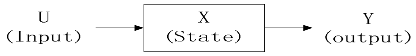
Figure 3: Compatible To Current Simulation Models

5.5 Simulation Engine Design

Simulation engine is a simulation control program that drives simulation files according to simulation process operation. The drive modes are different for continuous and discrete simulation.

5.5.1 Discrete event simulation of continuous system

(1) Discrete event model of continuous system



$$y = f_o(t, x, u) \tag{1}$$

$$x = f_d(t, x, u) \tag{2}$$

$$x_{d_{k+1}} = f_u(t, x, u) \tag{3}$$

Where,

$$x = x_c + x_d \tag{4}$$

(2) Discrete event simulation process of continuous system

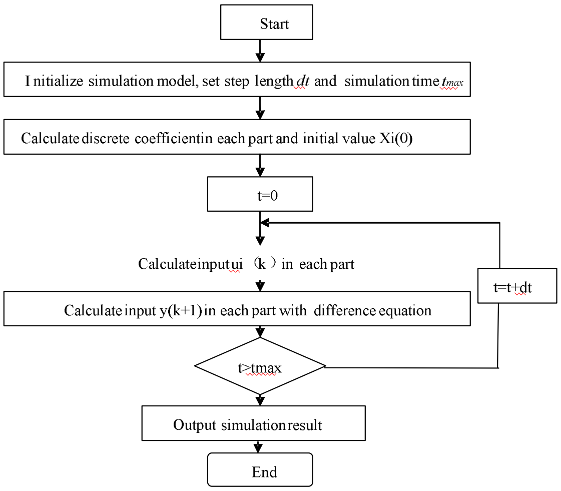

Start
Initialize simulation model, set step length dt and simulation time t_{max}
Calculate discrete coefficientin each part and initial value Xi(0)
t=0
Calculateinputui (k) in each part
t=t+dt
Calculate input y(k+1) in each part with difference equation
t>tmax
Output simulation result
End

Figure 4: Digital Simulations With Discrete Analog Method Flowchart Of Structure Diagram

5.5.2 Discrete event system simulation process

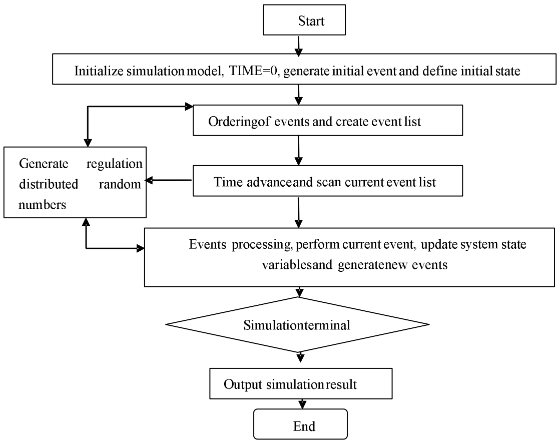

Start
Initialize simulation model, TIME=0, generate initial event and define initial state
Orderingof events and create event list
Generate regulation distributed random numbers
Time advanceand scan current event list
Events processing, perform current event, update system state variablesand generatenew events
Simulationterminal
Output simulation result
End

Figure 5: Simulation Mechanism Flowchart Of Discrete Event System

5.5.3 Drive process design of simulation engine

There are great differences between continuous system simulation and discrete event simulation in simulation method. The continuous system usually adopts numerical integration or discrete analog method and uses differential equation, state equation or difference equation to describe the system. It also concerns the instantaneous state of dynamic system when the simulation result is obtained. However, discrete event simulation usually takes Monte Carlo Method. Its core is discrete event’ s arrangement and processing and simulation clock step. Although their simulation methods are different, they both belong to clock step driving mode in simulation driving process. If the events scan and generation mechanism of discrete events are described by system input and output, the simulation operation in every clock step of both systems can be described by structure diagram. Therefore, a common driving process of simulation engine can be designed, which can meet both system needs.

6. CONCLUSIONS AND FUTURE WORK

The characteristics of distributed co-simulation platform based on .NET are: on one hand, it enhances the generic of simulation system by way of supporting various collaborations, on the other hand, it improves the ability of extendable and flexibility of system. It is easy to integrate current application and develop new application. The key technologies discussed in this paper have been applied in the project of Communication System. Co-simulation Platform, in the next work we will extend the applying domain of this platform and study the key technologies of designing generic co-simulation platform based on WWW browser by the tool of ASP.NET.

ACKNOWLEDGMENTS

This work was supported by The State Key Laboratory of Complex Electromagnetic Environmental Infects on Electronics & Information System, China.

REFRENCES:

[1] G. Borriello, R. Want, “Embedded Computation Meets the World Wide Web”, *Communications of the ACM*, Vol.43, No.5, 2000, pp. 255-258.

[2] F. Kuhl, R. Weatherly, J. Dahmann, “Creating Computer Simulation Systems: An Introduction to the High Level Architecture”, Prentice Hall, *MITRE Corporation*, Vol 10, 2000, PP. 233-235.

[3] D. Wildt, F. R. Wagner, “Adapting Simulation Environments to HLA: Discussion and Case Study”, *Proceedings of the European Simulation Multiconference, Prague, Czech Republic*, Vol.6, 2001, pp.120-123.

[4] Wang Jiangyun, Wang Xingren, Jia Rongzhen, “Architecture of Collaborative Simulation Environment”, *Journal of System Simulation*, Vol. 11, 2001, pp. 687-489.

[5] Wang Jiangyun, Wang Xingren, Chai Xundong, “Research and Implementation of COSIM Run-time Management Platform”, Vol. 5, 2003, pp. 585-587.

[6] Sun Zhixin, Wang Ruchuan, “Research and Implementation of Distribute Virtual Simulation Platform Based on Component”, *Journal of System Simulation*, Vol. 11, 2001, pp. 417-418.

[7] M. S. Hasan, H. Yu, A. Carrigton, T.C.Yang, “Co-simulation of wireless networked control systems over mobile ad hoc network using SIMULINK and OPNET”, *IET Communications*, Vol. 3, No. 8, 2009, pp. 1297-1310.

[8] X. Xian, W. Shi, H. Huang, “Comparison of OMNeT++ and other simulator for WSN simulation”, *IEEE Conference on Industrial Electronics and Applications*, Jun. 3-5, 2008, pp. 1439-1443.