III INTERNATIONAL CONFERENCE
TRANSPORT SYSTEMS TELEMATICS TST'03

ZESZYTY NAUKOWE
POLITECHNIKI ŚLĄSKIEJ 2003
TRANSPORT z.51, nr kol. 1608

discrete event simulation, computational Grid, virtual application, Web service, transport processes

Ewa OCHMAŃSKA[1]

COMPUTER SIMULATION OF TRANSPORT PROCESSES ON THE WEB PLATFORM

A framework for discrete event simulation of transport processes, based on Open Grid Services Infrastructure and accessible via Web portal, is presented in the paper. An object-oriented and data-driven nature of underlying simulation system facilitates its Web-based, distributed implementation using Java, XML and Web Services technologies. Grid infrastructure can provide controlled, secure and transparent access to the virtual application composed of distributed information and computational resources, by means of standard Web browsers. Simulation models for various categories of processes specific to different transport modes can be collectively defined, executed and analyzed. The semantic scope of simulation environment can be cooperatively extended through introducing new schemas for particular categories of transport processes.

KOMPUTEROWA SYMULACJA PROCESÓW TRANSPORTOWYCH W SIECI WWW

W referacie prezentowane są ramy systemu symulacji zdarzeń dyskretnych dla procesów transportowych, opartego na Otwartej Infrastrukturze Usług Gridu (OGSI) oraz dostępnego za pośrednictwem portalu WWW. Zorientowany obiektowo i sterowany danymi charakter systemu symulacyjnego ułatwia jego rozproszoną implementację w sieci Internet z wykorzystaniem technologii Java, XML i usług WWW. Infrastruktura Gridu może zagwarantować transparentny, bezpieczny i kontrolowany dostęp, przez standardowe przeglądarki internetowe, do wirtualnej aplikacji złożonej z rozproszonych zasobów informacyjnych i obliczeniowych. Proponowane rozwiązanie pozwala na grupowe definiowanie, wykonywanie i analizowanie modeli symulacyjnych dla różnych kategorii procesów specyficznych dla różnych gałęzi transportu. Semantyczny zakres środowiska symulacji może być rozszerzany przez kooperatywne wprowadzanie nowych schematów dla poszczególnych kategorii procesów transportowych.

1. INTRODUCTION

Several WWW technologies are emerging, leading to quite new possibilities of sharing various computational resources available through Internet and of organizing distributed cooperative application frameworks for problem solving. Two main approaches being developed in the recent years are computational Grids of scientific communities [1] and Web

[1] Faculty of Transport, Warsaw University of Technology, Koszykowa 75, 00-662 Warszawa, och@it.pw.edu.pl

services supported by commercial organizations [2]. The first of them offers intermediate programming layer for controlled but transparent access to distributed computing resources, another one introduces Web standards to automate communication between distributed service providers. Both approaches lead to new architectures of virtual applications, organized around specialized Web portals. The concept of Web services has been introduced to Grid environments resulting in Open Grid Service Architecture (OGSA) [3].

This kind of computing infrastructure is adopted in the paper as framework for defining computer discrete event simulation models of transport processes and executing them on the Web platform. Proposed simulation environment bases on a particular method for defining, constructing and executing simulation models. The method described in Section 2 has been previously applied in simulation models of technological and control processes at railway stations [4, 5] as well as in models of logistic processes for transport and distribution [6]. Process models are defined according to predefined schemas, including semantic classes of elements and "construction rules" for process categories. General, object-oriented and data-driven methodical principles permit to embed described modeling approach in the structure of Web portal, providing common access to the virtual application and its data resources.

Simulation portal will supply user with several functions concerning definition of dynamic process models for particular transport activities, specification and execution of simulation jobs as well as visualization and analysis of results. Special functions, including definition of process schemas, will allow collective developing and deploying of models for new semantic categories of processes, applying to various transport modes. Section 3 situates particular functions concerned with simulation research in the context of Web- and service-oriented architecture, using XML [7] and Java [8] based technologies to integrate distributed data and program components in collaborative simulation environment.

Presented project concepts, sketched in Section 4, assume implementing the simulation environment as a virtual application on top of Grid infrastructure, accessed via specialized simulation portal. Client-side activities are localized in the frame of Web browser. Web server provides user with GUI implemented in XML/XSL technology in connection with Java based scripting tools [9], transforming XML Schemas of processes in adaptable creators for defining valid models and specifying well formulated tasks for simulation research.

Final remarks concern possibilities and scope of collaborative use of simulation portal. Described Web-based environment for simulation modeling can be used as open cooperation platform in investigating mutually dependent processes in transport domain. Equipped with proper educative examples it also can serve as a support for teaching [10].

2. METHODOLOGY OF SIMULATION MODELING

2.1 OBJECT-ORIENTED IMPLEMENTATION OF MODELS

A simulation model is defined as a net structure consisting of any number of concurrent processes. A process is modeled by Petri net i.e. bi-graph with two disjoint subsets of nodes: static places containing tokens, and dynamic transitions moving tokens between places. Transitions are ordered by time in bi-directional list representing queue of events.

In an initial state of simulation model, some places contain some tokens representing processing and processed entities. During simulation transitions perform actions, corresponding to events, changing states of simulated processes by consuming and producing tokens. Detailed description of modeling Petri net tool, with semantic extensions introduced for simulation purpose and with an outline of implementation, can be found in [11].

Generic implementation of a simulation model contains four base object classes of components. Specific models are composed of semantic subclasses of objects derived from the base classes. The base object classes and their semantic extensions are resumed in Tab.1.

Table1

Object classes of model components

Object class	Meaning of base class	Extensions in derived classes	Examples of semantic interpretation
Token	Processed / processing entity	Data structure describing properties and actual state of represented entity	Vehicle, worker, single way or track, load unit
Place	Container for tokens (i.e. entities participating in process)	Type of contained entities	Parking place for vehicles, group of workers, section of route, warehouse building
Transition	Event changing process state by consuming and producing (using) entities	Action performing state changes related to particular event; Condition enabling event occurrence	Unloading transport vehicle, moving vehicle between two points of route
Process	Controlling succession of events; communication with other processes	Control rules specific for process semantics; specification of external input/output classes of places	Transferring goods in a net of transport nodes; serving arriving trains at a station

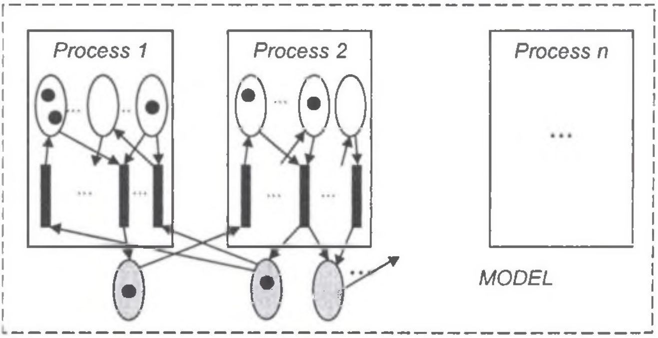

Fig.1. An exemplary fragment of a net model

Fig. 1 shows mutual relationships between four base classes of components on a piece of imaginary model. Tokens, places and transitions are represented on the figure by small black circles, ellipses and bars, respectively. Rectangles represent processes, which communicate by external input/output places (distinguished by gray color on the figure). The synchronization issues, arising in the case of executing processes in separate (possibly distributed) threads, can be resolved by several known approaches, e.g. [12]. An additional supervisory component of the model object class may be introduced for that purpose.

2.2 DATA-DRIVEN MODEL CONSTRUCTION

Discrete event simulation consists in execution of abstract programmatic models imitating behavior of real word systems, performing particular processes. Program representation of dynamic net model is constructed of object instances of semantic components, derived from four base classes, with extensions proper to the categories of modeled processes.

Program instantiation of a model is constructed following previously elaborated definition, specifying object classes of all model components and their mutual connections. Defining a model is in turn guided by schemas for particular categories of processes taking part in model performance. A part of model representing process of some category, with particular structure, dynamics and semantics, should be defined according to the process schema specifying following construction rules:

- a list of semantic object subclasses derived from places, which may or must appear in process model
- definitions of data structures, assigned to semantic classes of tokens, corresponding to particular semantic subclasses of places
- a list of semantic object subclasses derived from transitions which may or must appear in process model; each semantic subclass of transitions is described by semantic subclasses of its (obligatory or optional) input and output places
- composition of process model given by:
    - ❑ relations partially ordering semantic subclasses of transitions according to processing flow (transition subclasses in ordered sequences may be obligatory or optional)
    - ❑ subsets of semantic subclasses of external input/output places, containing tokens of semantic subclasses required as input to or produced as output from a process.

Two processes can cooperate if they are linked through a place of common semantic subclass, specified as external output from one of them and external input to another.

Simulation program, instantiating defined model as an interrelated structure of object components, is then executed according to a task specification, including parameters and initial states of processes defined appropriately for planned simulation experiment. Specifying simulation task refers to model definition and must conform with applied process schemas, in particular with definitions of data structures assigned to semantic tokens.

Process schemas, model definitions and task specifications are organized in textual data sets, which drive program modules performing preparatory phases of simulation, leading to initialized model instantiation ready to execute simulated processes, as it is shown on Fig. 2.

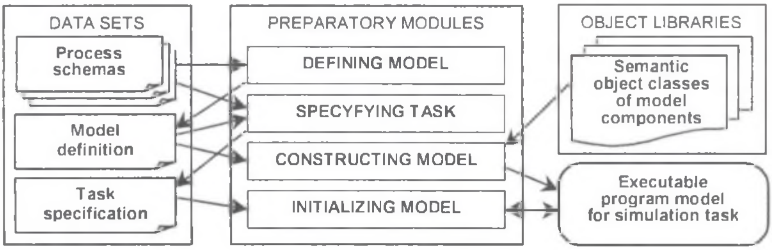

Fig.2. Data-driven definition, construction and initialisation of simulation model

All the preparatory program modules of simulation system use (and some of them produce) textual data sets driving their execution. Besides, the module constructing executable program model following its textual definition makes use of object libraries of semantic model components, corresponding to modeled process categories.

3. WEB-BASED IMPLEMENTATION OF SIMULATION SYSTEM

3.1 MODELING PRINCIPLES AND INTERNET TECHNOLOGIES

Object-oriented and data-driven principles of building simulation programs facilitate implementation of the projected simulation environment in the Web-based context by means of the two most widespread Internet technologies: Java for programs and XML for data.

Java, the object-oriented language developed by Sun Microsystems, gained great popularity in distributed Internet applications thanks to its Web-oriented specific, platform independence, libraries of API tools and open source license. The net structure of semantic subclasses of simulation model components can be naturally programmed in the object-oriented Java language. Four base model components and their semantic extensions will be implemented as Java object classes and subclasses, grouped in general and specialized packages. Packages corresponding to various categories of processes in different transport modes can be distributed in the network. Program components needed to construct defined model will be remotely accessed by standard Internet protocols.

XML (*eXtensible Markup Language*), the Web standard for self-describing textual data format, is widely used by Web applications as well as by several high level communication protocols. Data-driven preparatory simulation phases in described simulation system may be easily implemented by XML-related standards. Instantiation and initialization of simulation program can be driven by XML documents containing model definition and task specification, with use of XML parsers accessible in open libraries of Java tools.

According to earlier described modeling principle, semantic validity of model definition is assured by its conformity to process schema. Semantics of process schemas can be expressed by standard verification mechanism of XML Schemas. Casual coincidence of names appears meaningful. The emerging Web standard of XML Schemas - with its nested, hierarchical structure, attributes and mutual references - is equipped with very appropriate terms to express the contents of process schema. The second part of XML Schema specification: Datatypes, provides comfortable way to describe data structures assigned to semantic subclasses of places. Process schema, containing construction rules for its simulation models, as described in subsection 2.2, can be implemented as XML Schema comprising model composition and data types assigned to tokens.

XML Schemas can then be used for defining valid simulation models of particular process categories in XML format and for specifying consistent simulation tasks. Moreover, together with XML document transformation tools and client-side programming techniques they permit to build context-dependent graphic user interface for defining model and specifying simulation jobs. Process XML Schemas can be treated by XSL (*eXtensible Stylesheet Language*) standard transformation techniques for XML documents in order to implement individual context-sensitive creators for building, initializing and executing simulation models for particular processes.

3.2 FUNCTIONAL PHASES OF SIMULATION

Model definition and execution are crucial in computer simulation, however several complementary phases are also necessary and equally important for making use of simulation research methodology. Input data for an experiment (usually consisting of several simulation passes) have to be specified in terms of initial states for modeled processes. Program representation of defined model has to be created and initialized with input data before being executed. Output data resulting from simulation passes demand analysis and visualization.

Each of mentioned above functional phases of simulation requires software configuration for processing specific input data, including user activity, to produce specific results. This is resumed in Tab. 2 in the context of previously described Web tools for defining and implementing simulation models.

Table 2

Functional phases of simulation and their I/O data

Phase	User activity	Input	Output
Defining simulation model	Choosing process categories; Building model configuration of semantic components	XML Shemas for process categories	XML definition of simulation model
Specifying simulation task	Defining model parameters; Assigning tokens with data structures for initial states	XML model definition & XML Shemas for processes (data structures)	XML specification of simulation passes
Synthesizing simulation model	Demanding execution of specified simulation task	XML model definition XML task specification	Java program with created and initialized net model structure of components
Running simulation job	"	Initial state of programmatic model instantiation	Row result describing events recorded during simulation passes
Elaboration of simulation results	Specifying queries on simulation output data; Choosing presentation forms	Recorded simulation results	Analytical/synthetic views of simulation results and their visualization

4. OPEN SIMULATION ENVIRONMENT IN THE WEB

4.1 COMPUTATIONAL GRIDS & WEB SERVICES

In the last decade many efforts have been concentrated on elaborating tools for sharing various computational resources available through Internet. Two main approaches to organize Internet resources in consistent and safe working environments were emerging: computational Grids for e-Science and Web Services for e-Business.

The idea of computational Grids comes from scientific and engineering communities, aiming at sharing advanced computing resources among different institutions in fair and effective way. Grid environments serve as a platform for organizing well-controlled but transparent use of computing nodes, mass storage units, software, data archives, scientific apparatus and other distributed resources connected by Internet.

Grid systems base on intermediate programming layer called middleware, providing such services as advanced access control with single login to all needed computing resources, secure data transfer, resource management, monitoring and scheduling of computational jobs, information about actually accessible resources etc. Several Grid API's (*Application Programming Interfaces*) are available, also as the open source software, for implementing functions of Grid middleware.

Web Services approach was firstly introduced by commercial organizations (including MicroSoft and IBM) as a solution for business to business and customer to business Web applications. WSDL (*Web Service Description Language*) and other related technologies are supported and standardized by W3C (*World Wide Web Consortium*). Service-oriented architecture of distributed Web applications has been broadly accepted. It was adopted also by developers of Grid solutions, resulting in OGSA (*Open Grid Services Architecture*) standard. Grid services are defined in accordance with WSDL specification. On the other hand, Grid-type infrastructures are being introduces to commercial solutions.

The new service-oriented architectures lead towards virtual Web-based applications, consisting of mutually related services, using virtual pool of resources distributed in the Web. As it is shown on Fig. 3, a virtual application may be built of different kinds of service components: performing functions specific to the application, organizing access to the Grid infrastructure and providing other common Web services. This type of architecture is proposed in the following subsection as a framework for earlier described simulation environment, to be implemented as a virtual application embedded in specialized Web portal.

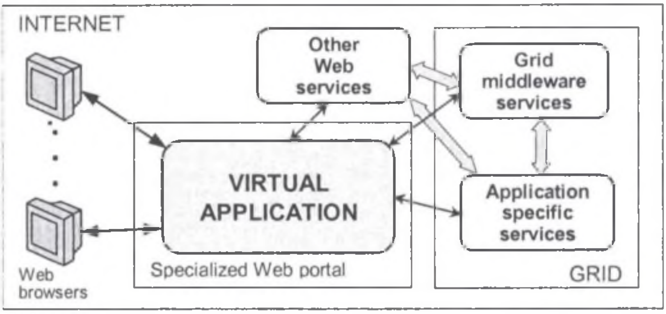

Fig.3. Service-oriented virtual application in the Grid context

4.2 VIRTUAL APPLICATION IN THE SIMULATION PORTAL

Presented project concepts lead to implement described simulation environment as a virtual application on top of grid infrastructure, accessed via specialized Web simulation portal. An outline of its functional architecture is sketched on Fig. 4.

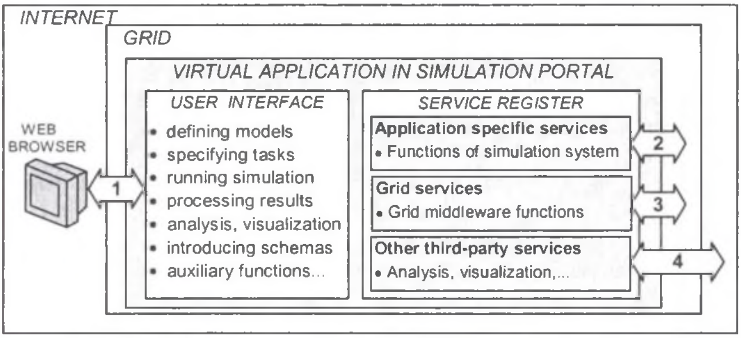

Fig.4. Functional structure of service-oriented virtual application in Web simulation portal
1 – XHTML/XML + scripts & applets; 2 – WSDL + application-specific protocols;
3 – WSDL-based Grid service protocols, 4 – WSDL + SOAP or other protocols

Client-side activities are localized in the frame of a Web browser and supported by XHTML/XML pages with built-in dynamics of Java scripts and applets. The Web server provides user with dynamic, context-dependent GUI for interacting with particular functions of simulation environment. XML/XSL techniques can be used on server-side in connection with Java based scripting tools like JSP, in order to transform process XML Schemas in adaptable creators for defining valid models and specifying well formulated tasks, as well as to produce interactive user interface for remaining phases of simulation experiment.

The server of simulation portal maintains a register of virtual application services with their access rules and localization of corresponding resources distributed in the Web. Access to services bases on WSDL in connection with appropriate communication protocols, depending on service kind, as is marked by white numbered arrows on the figure. SOAP *(Simple Object Access Protocol)* is widely used for communication with Web services.

Services of Grid middleware layer should organize controlled, single-login access to resources needed by virtual application services, including distributed simulation software, process schemas, data describing simulation models, tasks and results as well as computing nodes and storage. Model definitions can be archived, in order to be shared among cooperating groups of users. All kinds of resources related to application services may be situated anywhere in the Web.

5. FINAL REMARKS

The conception of adopting discrete event simulation modeling system, created earlier as a software tool for computer simulation of various processes in the domain of transport, leads to a Web and Grid service-oriented virtual application in specialized simulation portal. Proposed solution, exploiting new possibilities of resource sharing and cooperative Web-based technologies, can be used as collaborative simulation environment for creating and investigating advanced, composite models of mutually dependent processes in different parts of transport chain and in several modes of transport. On the other hand, the educative models of such processes placed in open simulation environment accessible via Web portal can serve as teaching support for transport faculties.

The simulation portal described in the paper will permit for cooperation in defining models, planning experiments and elaborating task specifications. Existing model definitions can be archived in model libraries and shared by several users. Related models can be composed in meta-models and then executed in multi-thread simulation.

Moreover, the Web-based simulation environment can be collectively evolved. Authorized expert users can introduce XML Schemas for new categories of processes to the open simulation system, along with correspondent Java libraries of semantic model components, thus extending semantic scope of virtual application.

BIBLIOGRAPHY

[1] Global Grid Forum, http://www.gridforum.org ; e-Science Programme, http://www.escience-grid.org.uk
[2] Web Services Activity, http://www.w3.org/2002/ws
[3] FOSTER I. et al, The Physiology of the Grid, An Open Grid Services Architecture for Distributed Systems Integration, http://www.gridforum.org/ogsi-wg
[4] OCHMAŃSKA E., Simulation of Serving Trains at Railway Freight Stations, Proceedings of the 13th European Simulation Multiconference, Warsaw 1999.
[5] OCHMAŃSKA E., WAWRZYŃSKI W., A Simulation Model of Transport Node Control System, Archives of Transport, vol.14, issue 4, Polish Academy of Science, Committee of Transport, Warsaw 2002.
[6] OCHMAŃSKA E., Object-oriented PN Models with Storage for Transport and Logistic Processes, Proceedings of the 9th European Simulation Symposium „Simulation in Industry", Passau 1997.
[7] Extensible Markup Language, http://www.w3.org/XML
[8] The Source for Java Technology, http://java.sun.com
[9] KOCHMER C., FRANDSEN E., Tworzenie aplikacji i usług WWW za pomocą JSP i XML, Wydawnictwo Helion, Gliwice 2002.
[10] MROCZEK K., FILASIAK R., XML w systemach zdalnego nauczania, Materiały z konferencji "Polski Internet Optyczny. Technologie, Usługi i Aplikacje. PIONIER 2001", Politechnika Poznańska 2001.
[11] OCHMAŃSKA E., An Implementation of Time and Control in Simulation Models of Transport Systems, II International Conference "Transport Systems Telematics", Katowice-Ustroń 7-9 November 2002. Politechnika Śląska, Zeszyty Naukowe Nr 1570, Transport z.45, Gliwice 2002.
[12] NICOL D., LIU J., Composite Synchronization in Parallel Discrete-Event Simulation, IEEE Transactions on Parallel and Distributed Systems, May 2002, Vol. 13, No. 5.

Reviewer: Ph. D. Stanisław Krawiec