MICROSIMULATION MODEL DEVELOPMENT ENVIRONMENTS

Istvan Molnar
School of Business, Department of Computer and Information Systems
Bloomsburg University of Pennsylvania,
Bloomsburg, Pennsylvania, 17815, U.S.A.
E-Mail: imolnar@bloomu.edu

KEYWORDS: Microsimulation, Socioeconomic applications, Microsimulation software, Web-services,

ABSTRACT

This paper describes some of the results of a search for a microsimulation software and analyzes new software technologies (among others Web-based application development), which can be applied to develop microsimulation models.

First, the application field and some of its major characteristics and requirements are introduced. After a short introduction and classification of microsimulation models, basic technological approaches for microsimulation model implementations are presented. Next, two major technologies, the database-oriented and the web-oriented approaches are discussed in detail. The two major solutions for realizing Web-services are J2EE and .NET. These two technologies are compared and their impact on microsimulation and related software technology shortly discussed. Final remarks conclude the paper.

INTRODUCTION

Microsimulation is a method able to handle complex socioeconomic systems by creating and studying a model that makes intensive use of the statistical data of the observed objects. These objects are the so-called *micro units* of the socioeconomic system; the *person*, the *family* or the *household*. The microsimulation models use simulation techniques in order to study the behavior of micro level units in time (see also Orcutt et al. 1961).

Microsimulation is generally accepted by decision-makers and widely used in Australia, Canada, Europe and the USA to prepare political decisions (see O'Donoghue 2001). Not just highly developed economies, but economies in transition also face many problems especially in demography, pension systems, health care, and taxation. Microsimulation can be a very useful tool to a model-based study of related problems and possible solutions.

Generally, two different microsimulation model classes were developed in order to build realistic models: *data-driven models* and *agent-based models*. Despite the different modeling approaches, both model classes handle model data and methods in a similar way; in both cases, significant amount of data must be analyzed and processed.

One of the most important technical problems of microsimulation model implementations is the integration and usage of different data sources available for microsimulation models. Historically, three different approaches have been developed:

- File processing approach (e.g., Heike et al. 1994)
- Database-oriented approach (e.g., Sauerbier 2002)
- Agent-oriented approach (e.g., Pryor et al. 1996)

These approaches use mainframe or PC technology and as such, are not portable and architecture neutral. In the 90-ies, new network-oriented technologies were developed in order to support applications (like model-based analysis) using heterogeneous hardware and/or software platforms. Nowadays, the development of networked multi-platform microsimulation applications is not just necessary but also technically possible; beyond networked data access distributed computing can also be realized.

In Hungary, after joining the European Union, more and more signs indicate an increasing demand for instruments of macroeconomic analysis and prediction, coupled with a tendency of more willingness to budgetary spending for microsimulation. There is an urgent need to find and/or develop advanced software tools, which support the achievement of the following goals:

- Network oriented data and model access.
- Distributed model execution.
- Multi-platform hardware and software solutions.
- Open standard based software solutions.
- Data and network security.
- User friendliness
- Efficiency

Proceedings 16th European Simulation Symposium
György Lipovszki, István Molnár © SCS Press, 2004
ISBN 1-56555-286-5(book) / ISBN 1-84233-106-x(CD)

MICROSIMULATION

*Microsimulation model*s have different data elements: *initial model data*, *intermediate* and/or *final simulation data;* all of these data are stored for further analysis. *Model behavior* in micro-simulation models is described using algorithms, which reflect the behavior processes of the micro units and represent their environment. Special care is taken to do the *data analysis* and the estimation of simulation *model parameters*. The microsimulation model is working in an *experimental framework* in order to study the effects of policy changes on the microsimulation model behavior. See Figure 1.

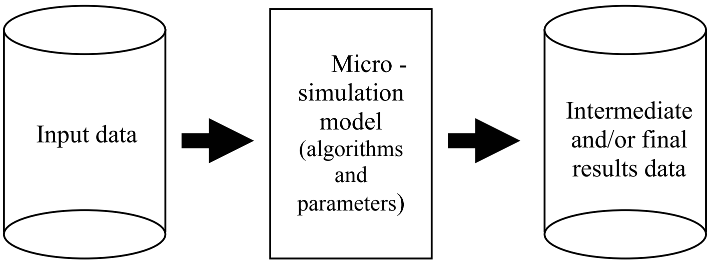

Figure 1: Microsimulation model

Input data, intermediate and/or final results' data are carefully analyzed and special techniques have been developed to improve data quality (e.g., imputing, merging, synthetic data). Model verification and validation also uses different methods and techniques (see e.g., Rubin 2004; Little and Rubin 2002; Schofield and Polette 1998)

TECHNOLOGICAL ALTERNATIVES

Considering the available network-oriented technologies, Web-enabled microsimulation models can be developed using different technologies. Following the IT industry main trends, two basic approaches can be distinguished':

- Database-oriented approach
- Web-service approach

Database-oriented Approach

The database-oriented approach is based on RDBMS technology — models are implemented and used in a RDB environment using different analytical tools and technological standards (incl. also Web technologies, mathematical and or statistical program packages and advanced user control and interface).

The network-oriented RDBMS provides a possibility to develop advanced microsimulation applications using architecture as depicted in Figure 2.

This architecture emphasizes data management and provides a multi-platform accessibility for microsimulation data over the network. Applications processing these data are not considered as "special," but rather as "usual" DB applications.

Several commercial software products are available that can take over the main burden of networked database management (e.g., Oracle Application Server 2003). Microsimulation algorithms can be developed using Java; data analysis and parameter estimation can be prepared by special mathematical and statistical software tools (e.g., SAS, SPSS).

The elements of this technology are widely available and well proven, industry-standard, vendor-neutral and platform-independent solutions are extensively used.

Web service Approach

Web service is a transition to service-oriented, component-based, distributed applications. Web services are applications implemented as Web-enable components with well defined interfaces, which offer certain functionality to clients via the Internet. Once deployed, Web services can be discovered, used/reused by consumers (clients, other services or applications) as building blocks via open industry-standard protocols. Web service architecture is built on open standard, vendor-neutral specifications. Services can be implemented in any programming language, deployed and then executed on any operating system or software platform.

The software architecture of Web services is presented in Figure 3.

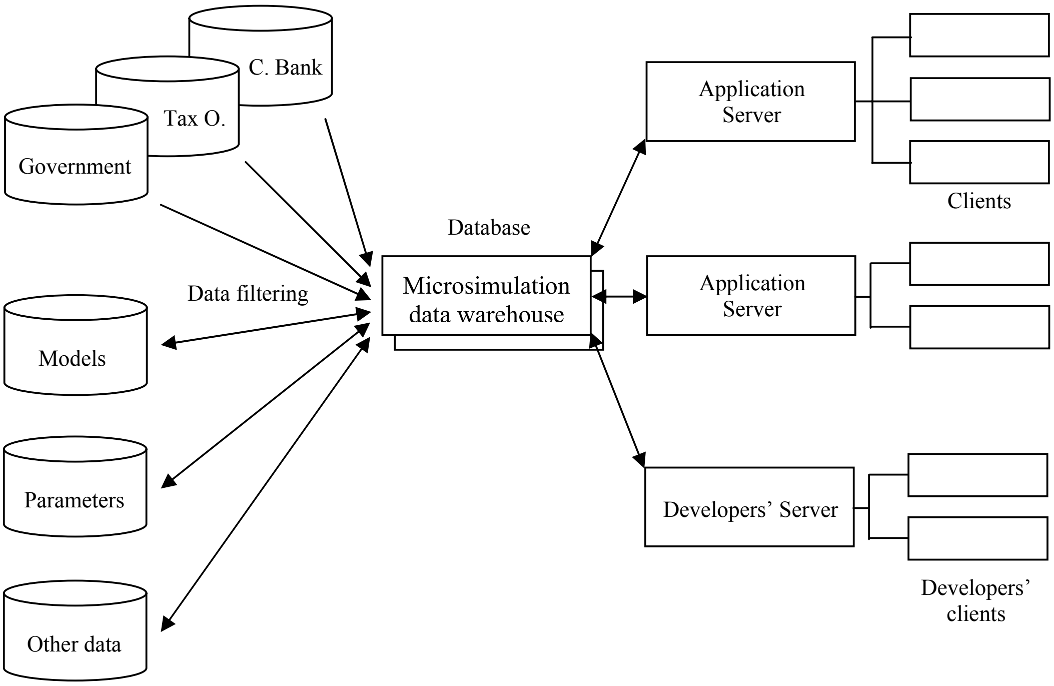

Figure 2: Database-oriented architecture

The model behind Web services is a loosely coupled architecture, consisting of different software components working together. Before invoking a service, consumers must first locate the application offering the service needed, discover the interface and then configure their software in a way that it is able to collaborate with the Web service.

Consuming Web services is based on open standards managed by broad consortia. The Universal Description Discovery and Integration (UDDI) is responsible for publishing, locating and binding Web services to consumer software. The service, requested by the consumer is determined by a contract between the service provider and the client, who will consume the service. The contract can be formulated using the Web Services Description Language (WSDL). Using the Simple Object Access Protocol (SOAP) and Hypertext Transfer Protocol (HTTP), the parties involved will agree upon a common message and protocol. The data interchange format, used during all communications, is also standardized; the Extensible Markup Language (XML) is used.

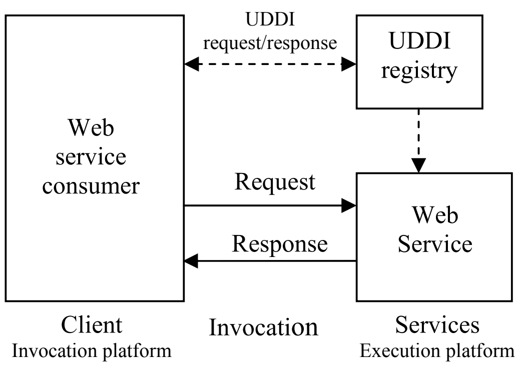

Figure 3: Web service architecture

Currently, two main platforms are used to develop Web services: Microsoft .NET and J2EE.

These technological developments will fundamentally question and/or extend previous network-oriented simulation technologies (e.g., Miller et al. 1998 and Miller et al. 2001). HLA-based solutions (e.g., Lantzsch et al. 1999) can further extend the possibilities of Web services and allow the user to develop large, multi-platform, network-oriented microsimulation models.

Unfortunately, sufficient experience is not available yet. The obvious advantages will certainly further attract simulation software developers in the future; consequently Web services will play a viable alternative for microsimulation software development as well.

Microsoft .NET Technology

Microsoft .NET is a new Windows platform for developing and deploying web services. The platform is a combination of new and old tools and approaches; .NET is optimized for XML and designed around XML-based Web Services. .NET extends the Visual Basic paradigm to all programming languages on the platform; any language can be used and they can also inter-operate. The most important .NET languages are Visual C# .NET and VB .NET. The platform incorporates a run-time environment called Common Language Runtime (CLR), which is similar to Java VM and referred to as "managed execution environment", which allows compiled programs to run on any platform. The framework also consists of other Web service related components like ASP.NET and Microsoft Internet Information Services (IIS)

Java Technology

Java technology includes the Java programming language, the runtime environment, the platform editions, and the application programming interfaces (APIs). The current platform editions are: J2ME, J2SE, and J2EE (Micro, Standard and Enterprise Editions), each of them with the Java language and its portable bytecode at the core. The major J2EE application server vendors are IBM, SUN, Borland, Oracle, and Macromedia. J2EE runs on several platforms, but only supports the Java language.

In a J2EE-based Web-service environment, different software elements, provided by different vendors, must work together, which can make the program development more difficult. Therefore development requires more professional attention, because process and code generation is not a highly automatic process, like in the case of .NET. To increase programming efficiency, it is strongly suggested to use an IDE (Integrated Development Environment), like IBM Web-sphere or SUN One.

One of the most critical elements of the development is the data communication between client and service provider (see Figure 3); both client and service provider must have the same elementary data type or have the map the given data types between Java and SOAP.

Comparison of .NET and Java Technologies

Comparing and evaluating application software, decision-makers usually examine the following criteria: TCO, software performance, development and other abilities (CGI Group Inc. 2002).

From the software developer point of view, the arguments and opinions are also very extensively discussed (see Benchmark comparison 2001 and The Middleware Company, 2003).

.NET was designed as "the platform for XML Web Services", while Web service technologies are not yet standardized in J2EE. Despite these facts, .NET and J2EE can, must and will coexist. Businesses or governmental institutions rejecting single vendor lock-in or preferring high level reliability, security or stability, might avoid .NET, but will miss out advantages, offered by the .NET platform.

CONCLUSIONS

The increased interest in microsimulation in Hungary creates an obligation to review the existing practices and get acquainted with the new technologies, which can be applied to new projects.

Converting traditional microsimulation models into Web-enabled ones provides for their effective and efficient use in modern integrated information systems. New technologies like Web service provide user-friendly and powerful tools for new model development.

Web-enabled microsimulation models can be used to develop Web-enabled model bases consisting of different type of microsimulation models and creating Web-enabled decision-support systems. The presented software-environments are generally available in contemporary IT environments.

REFERENCES

Benchmark comparison 2001. "Building XML-based Web Services in Microsoft .NET vs. IBM WebSphere 4.0". Revision 2.0, (Dec.).

CGI Group Inc. 2002. "Microsoft .NET or Java 2 Enterprise Edition: Is it just a question of platforms and languages?", White Paper, (Sept.)

Heike H.-D.; Beckmann K.; Kaufmann A.; Sauerbier T. 1994. „Der Darmstädter Mikro-Makro-Simulator– Modellierung, Software Architektur und Optimierung". In Proceedings: Faulbaum, F. (Ed.).

SoftStar'93–Advances in Statistical Software 4. Fischer. Stuttgart/New York, 161–169
Lantzsch, G.; Straßburger, S.; Urban, C. 1999. “HLA-basierte Kopplung der Simulations-systeme Simplex3 und SLX”. In Proceedings der Tagung Simulation und Visualisierung '99 der Otto-von-Guericke-Universität Magdeburg (Hrsg.: Deussen, O., Hinz, V., Lorenz, P.), Institut für Simulation und Graphik, (4.-5. März), Magdeburg.
Little, R. J. A. and Rubin, D. B. 2002. “Statistical analysis with missing data”. Wiley-Interscience. New York.
Miller, J.; Fishwick, P.; Taylor S.; Benjamin, P.; Szymanski B. 2001 "Research and Commercial Opportunities in Web-Based Simulation". Simulation Practice and Theory, Special Issue on Web-Based Simulation, Elsevier Science.
Miller, J.; Ge, Y.; Tao J. 1998. "Component-Based Simulation Environments: JSIM as a Case Study Using Java Beans," In Proceedings of the 1998 Winter Simulation Conference, Washington DC. (Dec.) IEEE, 373-381
O’Donoghue, C. 2001. “Dynamic micro-simulation: A methodological survey”. Brazilian Electronic Journal of Economics, vol.4, no. 2.
Orcutt, G.; Greenberger, M.; Korbel, J.; and Rivlin, A. 1961. „Microanalysis of socioeconomic systems: a simulation study”. Harper & Brothers. New York
Oracle Application Server 2003. “Oracle Application Server 10g J2EE and Web Services”. White Paper, ORACLE, (Sept.)
Oracle Application Server Portal 2003. “Oracle Application Server Portal 10g, Technical Overview”. White Paper, ORACLE, (Sept.)
Pryor, R., Basu, N., Quint T. 1996. “Development of Aspen: A microanalytic simulation model of the US economy”. Sandia Report #SAND96-0434, Sandia National Laboratories, Albuquerque, NM, (Febr.)
Rubin, D. B. 2004. “Multiple Imputation for Nonresponse in Surveys”. John Wiley & Sons. New York.
Sauerbier, Th. 2002. “UMDBS - A New Tool for Dynamic Microsimulation”. Journal of Artificial Societies and Social Simulation, vol. 5, no. 2
Schofield, D. – Polette, J. 1998. “A comparison of data merging methodologies for extending a microsimulation model”. NATSEM STINMOD Technical Paper No. 11. National Centre for Social and Economic Modelling, University of Canberra.
The Middleware Company, 2003. “J2EE and .NET (RELOADED) Yet Another Performance Case Study”. The Middleware Company, (June)

AUTHOR BIOGRAPHIES

Dr. Molnar was born in Budapest and educated at the Budapest University of Economic Sciences, where he received his MSc. and PhD. He has completed his postgraduate studies in Darmstadt, Germany and took part in different research projects in Germany as guest scientist in the 80-ies and 90-ies. In 1996 he has received his CSs. degree from the Hungarian Academy of Sciences. Currently, he is an Associate Professor at the Bloomsburg University of Pennsylvania. His main fields of interest are currently microsimulation, simulation optimization, simulation software technology, and simulation education. Dr. Molnar is a senior member of SCS International, member of the Editorial Board of SCS-European Publishing House and a former member of the European Council Board.