Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

WEB-BASED DIAGNOSIS OF MODEL SPECIFICATIONS

E. Joseph Derrick

Dawn Information Systems
Zi Jin Hua Yuan Tower #1 24A
15 Bin Shui Road, Hexi District
Tianjin 300061, CHINA

Department of Computer Science
Yenching Graduate Institute
197 Bei TuCheng Xi Lu
Beijing 100083, CHINA

ABSTRACT

Within the domain of Verification, Validation and Accreditation (VV&A) research, the automated diagnosis of model specifications has received little attention. As part of the limited effort focused on this important area, the Condition Specification represents one of the most promising model specification forms that is amenable to diagnosis. In addition, web-based technologies now offer exciting new possibilities to extend our knowledge in this and other areas of simulation research. This paper introduces the *SimDOG* Project: the development of prototype tools, written in Java, for the on-line and off-line (standalone) diagnosis of graph-based representations of Condition Specifications.

1 BACKGROUND AND MOTIVATION

A recent issue of IEEE *Computer* focuses on "Fighting Complexity in Computer Systems." The guest editor, Alexander Stoyen, recalls an interesting quote by David Parnas in his keynote address to the 1995 IEEE International Conference on Engineering of Complex Computer Systems. Parnas said, "Complexity is not a goal; I don't want to be remembered as an engineer of complex systems." Stoyen states that advances in computer hardware (bigger, better, faster) have ironically led to a problematic increase in the complexity of high-level software applications (Stoyen 1997). He further notes that the "lack of scalable methodologies for construction of large-scale, 'complex' systems and tools that support such methodologies is both well documented and chronic." Fortunately, this issue has generally not been overlooked by the simulation community. A considerable amount of research and related discussion has been focused in the last decade on simulation model development environments (Balci and Nance 1992; Paul and Hlupic 1994) including their standardization (Tanir and Sevinc 1994). A key component of a model development environment is *the model specification*, created under the influence of a guiding *conceptual framework* (Derrick et al. 1998). The development environment and its set of automated tools help modelers handle model complexity. In supporting this notion, it can be argued that the specification itself ought to possess several characteristics (Balci and Nance 1987): (1) *domain independence* (providing general applicability to simulation problem domains), (2) *automatically translatable* (relieving the modeler of the burden of low-level programming and implementation, thus enabling a focus on the model design, testing, and maintenance at the specification level), and (3) *analyzable* (assisting in model verification at an early stage, promoting effective verification and validation). This third and last characteristic has unfortunately received very little research attention and yet offers savings in project development time and costs. In fact, the larger umbrella of model verification and validation is often described as either neglected (Tanir and Sevinc 1994) or even worse, in danger of being forgotten (Paul 1998).

The Condition Specification (CS) (Overstreet 1982) and Simulation Graphs (Schruben and Yucesan 1993) are among the few specification formalisms that have shown demonstrated promise and amenability to automated diagnostic techniques. This paper describes the *SimDOG* (**Sim**ulation **D**iagnostic **O**n-line **G**raph-based analysis tools) Project and its application of web-based technologies to create automated tools to display graph-based representations of Condition Specifications for meaningful interpretation, further diagnosis, and simplification. A brief description of the CS and its diagnostic capabilities is given. The web-based technologies and the perceived benefits of their use in this domain are reviewed. An overview of the *SimDOG* Project is provided which covers the project goals, computing environment, current status, and future work. Concluding remarks describe the challenge in this research area.

2 THE CONDITION SPECIFICATION

Briefly, the Condition Specification (CS) (Nance and Overstreet 1988; Overstreet et al. 1994) produces a model specification that can be analyzed to (1) detect certain types of errors in the specification, (2) assist in creating efficient model implementations, (3) create helpful model documentation, and (4) provide data and information which will potentially lead modelers to a better understanding of the system under consideration. The CS has extensive analytic and diagnostic capabilities that offer significant benefits to modelers in the areas of *analytical* (existence of certain properties), *comparative* (differences between model representations), and *informative* (extraction or derivation of characteristics) diagnostic assistance. A variety of directed graph and matrix structures are derivable which effectively include the embedded relationships among model attributes and from which the cause and effect between these attributes can be readily determined. Analysis of these structures reveal information regarding attributes (utilization, classification, initialization, completeness, and consistency), the strength of relationships between equivalent conditions and the associated actions (called “cohesion”), and model components (connectedness, accessibility). This information can be extremely useful to modelers during the model development phases (e.g., measuring model complexity, simplifying model representations). Studies have been completed on the direct execution of CS forms for both sequential and parallel execution (Page 1994). Continuing research focuses on maximizing the utility of CS diagnostic capabilities (e.g., simplification techniques to recognize and eliminate redundancies (Nance, Overstreet, and Page 1996)).

3 USING WEB-BASED TECHNOLOGIES

The prototype tools of the *SimDOG* Project use the Java programming language as their basis. The following native Java features and their benefits are highlighted:

- *Portability*. Multi-platform compatibility is possible. Broad availability and access to project results within and across organizations (via intranet, extranet, or Internet) saves time and costs and supports collaboration.

- *Multi-threading and Client-server functionality*. Multithreaded server architectures enable the server side management of many simultaneous connections with many clients. This enhances the performance characteristics of servers dedicated to specification analysis tasks.

- *Enterprise APIs (Application Programming Interfaces)*. The JDBC (Java Data Base Connectivity, RMI (Remote Method Invocation), and Security packages (Flanagan 1997) assist development of project database facilities, distributed applications, and file and object access/manipulation.

- *New Version 1*.2 (Beta) *Features*. Using the new concepts of “permission” and “policy”, the Security API package offers fine-grain, highly configurable, flexible, and extensible access control for read-write-connect access to a particular resource (a specified file or directory, a given host and port, etc.). This covers all code written in the Java programming language, including applets, applications, beans, and servlets. Custom socket types support RMI over a secure transport such as SSL(Secure Socket Layer) (Sun 1998).

4 PROJECT OVERVIEW

The *SimDOG* Project goals are to (1) port existing diagnostic techniques as discussed above to Java-based tools, (2) provide new displays of graphical structures using native Java primitives (Java 2D and 3D APIs) and using other web-based technologies such as VRML, (3) maximize modeler interaction with the data and displays via client browsers, and (4) distribute the computationally intensive tasks to server-side while maintaining acceptable network performance characteristics between server and client.

In a typical *SimDOG* session, the modeler connects to the URL (Uniform Resource Locator) of a *trusted* Diagnostic Analysis Data (DAD) Server using his/her favorite Java-enabled client browser (see Figure 1.) This server could be on the modeler’s own machine, within the local intranet, remote extranet, or Internet. By use of a form fill-in, the modeler provides the file location (or possibly the URL) of a CS (in ASCII form) and gives permission for access to client resources. The DAD server releases control applets (e.g., initialization applet, display applet, etc.) for client functions as requested by the client. The first applet obtains access to client CS data, submits the data to the DAD server, and initializes the interactive interface between client and server. A set of server side Java programs parse the data, store resulting data in a database, and initiate additional applets which will visually display results to the modeler and provide interactive/ data manipulation features.

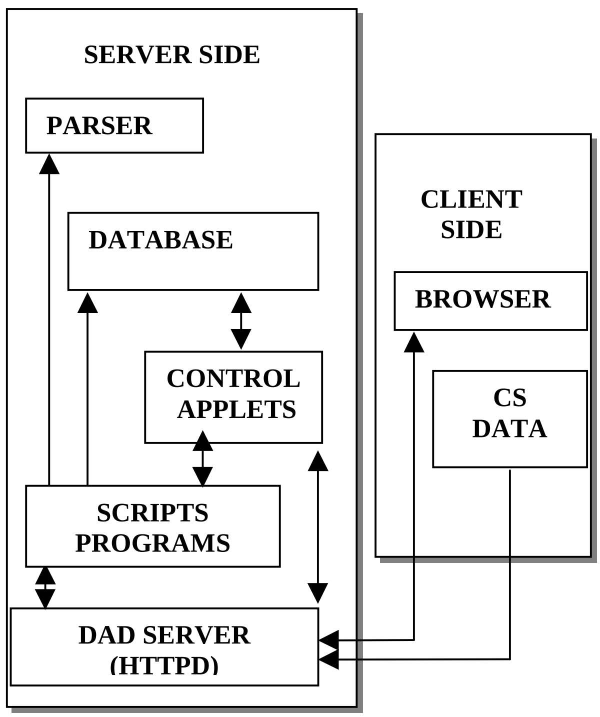

Figure 1**:** *SimDOG* Environment

The project computing environment consists of two Pentium Pro machines, one each for client and server side functions. Each machine is dual-boot (Linux-- Redhat 4.2 with kernel version 2.0.30, and Windows 95). Initial server side work is being done under Linux. However, due to potential future difficulties with the complex porting effort of the JDK beta version 1.2 to Linux, some later development may need to be shifted to another compatible environment (e.g., Win95). With the current dual-boot setup, not only is the primary Linux (client) to Linux (server) configuration testable, but also Win95 (client) to Linux (Server) and Win95 (client) to Win95 (server) combinations can be tested as tool development progresses and as Win95 compatible server-side components are developed. The components in Figure 1 and their current status include:

- *Parser*. The CS is parsed using Linux native tools flex and byacc. Parser development is underway and will be completed by August 1998. The initial use of flex and byacc allows work to proceed without waiting for development of a Java-based parser.

- *Database*. The mSQL version 2.0.3 database engine (Hughes 1998) is used.

- *DAD Server*. The httpd server is an Apache Server running various scripts and storing the control applets to coordinate server-side functions.

- *Database connectivity with the server* and limited applet communications (access to client side data, simple client side interface, and basic display of graphical representations) should be completed by November 1998.

Demonstration of feasibility of concept is projected for early first quarter 1999. An offline version of the toolset that facilitates standalone accomplishment of diagnostic activities is also proposed for a future date.

The *SimDOG* Project is intended to be a platform for ongoing and future research and development, not an overnight “quick-fix” implementation of CS diagnostic techniques. It will slowly evolve and mature. Successor tools could potentially integrate with other VV&A web-based services or within broader enterprise modeling solutions. The *SimDOG* Project assumes the pre-existence of a CS for input to its toolset. We believe this research will stimulate investigation into the improvement and adaptation of higher level conceptual frameworks for producing a CS or other promising specification formalisms.

5 CONCLUDING REMARKS

There is a documented need for research in the area of model verification and validation. Fundamental requirements continue to be overlooked; with the exception of a few areas (e.g., Condition Specification, Simulation Graphs, etc.), the diagnosis and analysis of the model specifications (early in the simulation study life cycle) receives little attention. The gains that have been achieved need to be realized in the development of research or model development platforms which will link or transform the formal methodologies (which possess diagnostic capability) to implementation level and usable tools. The *SimDOG* Project attempts to gain measurable progress in this transformation and generate renewed interest in this critical research area. In the same manner that today’s webmasters submit HTML code to web-based HTML validation services (WebTech 1998), might we perhaps (in the not-too-distant future) find that model developers are also able to submit model specifications to trusted local or remote analysis servers, thereby strengthening awareness, dependence, and rightful reliance on VV&A techniques?

ACKNOWLEDGMENTS

In support of this project, the cooperation and collaborative assistance from and between Carnegie Mellon University (Pentium computer equipment donation from Professor

John Shen), Professional and Educational Services International (Dr. Kenneth Chin), Yenching Graduate Institute (Professor Li Weinan), and Dawn Information Systems (Dr. Alan Sik, Managing Director) are gratefully acknowledged.

REFERENCES

Balci, O., and R. E. Nance. 1987. Simulation model development environments: A research prototype. *Journal of the Operational Research Society* 38 (8): 753-763.

Balci, O., and R. E. Nance. 1992. The simulation model development environment: An overview. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson, 726-736. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Derrick, E. J., O. Balci, R. E. Nance, and H. C. Shen. 1998. Conceptual frameworks for discrete event simulation modeling. Work in progress.

Flanagan, D. 1997. *Java Examples in a Nutshell*. O'Reilly and Associates, Sebastopol, CA, 99.

Hughes, D. J. 1988. Mini-SQL. Available online from **http://hughes.com.au/software/msql2/current.htm** [accessed April 1998].

Nance, R. E., and C. M. Overstreet. 1988. Diagnostic assistance using digraph representations of discrete event simulation model specifications. *Transactions of the Society for Computer Simulation* 4 (1): 33-57.

Nance, R. E., C. M. Overstree, and E. H. Page. 1996. Redundancy in model representation: A blessing or a curse? In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, 701-707. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Overstreet, C. M. 1982. Model specification and analysis for discrete-event simulation. Ph. D. Dissertation, Department of Computer Science, Virginia Tech, Blacksburg, Virginia.

Overstreet, C. M., E. H. Page, and R. E. Nance. 1994. Model diagnosis using the condition specification: From conceptualization to implementation. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, M. S. Manivannan, D. A.Sadowski, and A. F. Seila, 566-573. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Page, E. H. 1994. Simulation modeling methodology: Principles and etiology of decision support. Ph. D. Dissertation, Department of Computer Science, Virginia Tech, Blacksburg,Virginia.

Paul, R. J., and V. Hlupic. 1994. The CASM environment revisited. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, M. S. Manivannan, D. A.Sadowski, and A. F. Seila, 641-648. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Paul, R. J. 1998. Web-based simulation: Whither we wander? Panel session: The modeling methodological impacts of web-based simulation, Moderator: E. Page; Panelists: A. Buss, P.A. Fishwick, K. J. Healy, R. E. Nance, R. J. Paul, *1998 International Conference on Web-Based Modeling and Simulation*. Available on line from **http://www.cise.ufl.edu/~fishwick/webconf.html** [accessed April 1998].

Schruben, L., and E. Yucesan. 1993. Complexity of simulation models: A graph theoretic approach. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G. W. Evans, M. Molleghasemi, E. C. Russell, and W. E. Biles, 641-649. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Stoyen, A. D. 1997. Fighting complexity in computer systems: Guest editor's introduction, *Computer* 30 (8): 47, 48.

Sun Microsystems 1998. Java Development Kit Version 1.2 New feature summary . Available online from **http://java.sun.com/products/jdk/1.2 /docs/index.html** [accessed April 1998].

Tanir, O., and S. Sevinc. 1994. Defining Requirements for a Standard Simulation Environment, *Computer* 27 (2): 28-34.

WebTechs 1998. HTML Validation Services. Available online from **http://www.webtechs.com/html-val-svc/** [accessed April 1998].

AUTHOR BIOGRAPHY

E. JOSEPH DERRICK is a Partner and the Director of Research and Development for Dawn Information Systems, a software and internet services company with headquarters in Tianjin, China. He is also an Adjunct Associate Professor of Computer Science at Yenching Graduate Institute in Beijing, China. He received his B.S. degree at the U.S. Naval Academy in 1974 and M.S. and Ph.D. degrees at Virginia Tech in 1988 and 1992. He performed post-doctoral research in 1994-95 at Hong Kong University of Science and Technology in Computer Integrated Manufacturing and Flexible Manufacturing Systems. His current research interests focus on software engineering and web-based simulation and modeling. Prior to joining Dawn, he was a consultant to several multi-national companies in Beijing in a number of areas: systems integration, network and computer systems administration, software development, and Web design services. Dr. Derrick is a member of ACM and IEEE CS.