THE DESIGN OF AN XML-BASED MODEL DESCRIPTION LANGUAGE FOR WIRELESS AD-HOC NETWORKS SIMULATION

by

Andrew W. Hallagan

A Proposal Submitted to the Honors Council

For Honors in the Department of Computer Science

January 23, 2011

Approved: ____________________________

L. Felipe Perrone
Thesis Advisor

____

Stephen Guattery
Chair, Department of Computer Science

1 Introduction

In a network, communication is essential. Business professionals construct networks of clients and salespeople by using agreed-upon *mediums* of communication (e.g., written mail, the telephone, face-to-face conversation) and following the corresponding *protocols* (e.g., the memorandum, the conference call, the board meeting). A computer network is no different.

We can classify computer networks into three sets: hardwired, wireless, and *ad-hoc*. The members of these networks are individual machines, perhaps laptop and desktop computers. These members are called *nodes*.

The nodes in a hardwired network are connected with wires, such as Ethernet cable. Messages are sent back and forth along this cable according to a fixed protocol. Computers that contain the right hardware and software to follow these standards can communicate with each other and with an internet server. Wireless networks work in much the same way; the medium of communication in this case is electromagnetic radio waves, and the communication protocol might be the "WiFi" standard.

Wireless ad-hoc networks are a third type of network whose implementation is not so straight-forward. In general, an ad-hoc network is one in which nodes can freely communicate with both a radio signal-transmitting internet server as well as with other nodes nearby. The idea is that if any particular node is unable to reach the internet server directly (in case it is out of range), it can communicate with nearby nodes which can relay messages to one another along a chain until the messages reach a node that is connected to the internet server.

The nature of ad-hoc networks makes them hard to control and to observe; the individual paths each message takes to reach the internet server changes as the nodes move around and the network topology changes. Additionally, these networks should be able to scale up in size, but as the size of the network grows to say, even a few hundred nodes, it is not feasible to enumerate and analyze each possible path that each message might take. A major difficulty with ad-hoc networks is that their protocols have to consider these mobility and scalability issues, which complicate the routing of messages.

Thus, it is virtually impossible to predict the behavior and performance of an ad-hoc network using analytical methods alone. Instead of an analytical study researchers could certainly attempt an empirical one, but as discussed in [1], creating a physical network for this kind of study is prohibitively expensive and time-consuming. An important and increasingly valuable tool in the study of wireless ad-hoc networks is *simulation*, which is the process using software to set up an imaginary network and observing how it behaves. Every simulation requires a *model description*, which contains information about the number of nodes involved, the way those nodes are moving about, how much bandwidth is being used, the number of access points, etc. The description is given to a simulator engine, which is a program that executes the simulation and reports the results of an experiment.

Researchers have a number of network simulators at their disposal, including *ns-3*, *ns-2*, *GloMoSim*, *OPNET*, *CSIM* and others [2]. Each uses a different language for model description, which makes it very difficult for researchers to share and reproduce their work across platforms. The differences in model description languages, compounded with the sheer complexity of the models being described have had a serious negative impact on the credibility of many past simulation studies. Researchers should be able to trust that the models their peers use and publish are just exactly what they say they are – nothing more and nothing less. As it is now, this is not always the case. A researcher may assume the parameter values he has manipulated and explicitly defined are the only ones which need to be published, while another researcher using these explicit parameter values and a different set of default values can end up with entirely different results. In their report on the state of network simulation credibility, Kurkowski et al. [2] identified simulation setup as the most often ignored experimental phase. With a flawed setup, entire experiments are rendered useless.

While Perrone et al. [3] support the continuing emergence of new tools and systems for network simulation, they argue that a concerted effort towards increasing the *credibility* of research produced with these tools is requisite before any reliable progress can be made. A central claim of [3] and [4] is that this credibility will come when simulators can provide complete automation of the experimental process. In my honors thesis work I will make significant steps towards the realization of this goal by developing ways to create composable network simulation models for the *ns-3* simulator that can be validated and transformed with Extensible Markup Language (XML) technologies.

2 Background

XML is a markup language specification, standardized by the World Wide Web Consortium (W3C). It uses a system of tags to define *elements* in a document. The specification allows users to define their own tags and what kind of contents a particular element can contain. Following these rules, it is possible to express complex relationships between elements. It lends itself quite naturally to describing relationships between components of a simulation model, for example. The W3C has published specifications for other technologies that can be used for analyzing and manipulating these XML documents.

With Professor Perrone, I have already participated in a full-credit independent study, a half-credit independent study, and an 8-week research experience. I studied the model description process and how it can be improved to bolster credibility. During this time I familiarized myself with the existing research on network simulation and XML.

XML has been investigated by a number of researchers interested in creating simulations that are more universal and portable. XML is an appealing language because it is complemented by a suite of W3C-endorsed and third-party technologies for editing, querying, validating and transforming XML documents. Rioux, Bernier and Laurendeau [5] present a technology-independent conceptual framework for describing simulation scenarios, validating those scenarios, and then converting those scenarios into software objects that can be used by a particular simulator engine. They show that XML and some of its associated tools (JAXB, XQuery, XSLT and Native XML Database) can be used to implement this framework. Because these tools are available today, the time spent developing such an implementation is greatly reduced.

Röhl and Uhrmacher [6] discuss XML-based model components and show that the declarative specification of XML provides easier database integration, user readability and development of graphical user interfaces. In addition, XML happens to be an excellent data exchange format, allowing researchers to import and export model definitions in a more independent way. A disadvantage of describing simulation models in XML is the problem of converting a declarative specification (XML) to an imperative one (i.e., an *ns-3*-specific C++ script). For this reason, it is important to maintain a separation of code describing the *composition* of a model and code describing the *execution* of that model in the simulator engine. The authors demonstrate an implementation of this framework as part of the James II simulation system.

3 Project Description

In order to improve the credibility of research conducted with the *ns-3* simulator I propose developing an XML-based model description language as well a set of tools supporting the validation and transformation of that description into *ns-3*-compatible C++ or Python code.

Initially, this will involve using a number of common model description scenarios and parameterizing them in XML. I will transform XML code describing the components and parameters of these descriptions into basic C++ or Python code using a templating mechanism. I do not anticipate this to be especially difficult, however it will be helpful for the *ns-3* community to have a starting point for integrating XML model descriptions into their framework.

Later, as part of my research, I will develop ways for models to be composed and verified from scratch. This is a much more ambitious goal which involves considerable research and implementation. Given the work I have completed with Professor Perrone already, I am familiar enough with the issues surrounding this problem to pursue it with an understanding of its complexity. It is in this work that my research will move beyond mere implementation for *ns-3* and into a larger study of XML validation and code generation techniques that, while tailored closely to *ns-3*, will still be applicable to many other fields in computer science research.

This work will fit into Professor Perrone's larger vision of a more robust and user-friendly *ns-3* architecture. He and other members of the *ns-3* development team at the University of Washington and the Georgia Institute of Technology have secured a four-year N.S.F. grant (CNS-0958142) in order to work full-time towards developing software frameworks to automate the experimental process within *ns-3*. Fellow student Bryan Ward and I attended the SIMUTools 2010 research conference in Malaga, Spain last spring in order to present a poster outlining our preliminary thoughts on how to implement these frameworks [7]. The software components will generate scenarios, perform validation of simulation models, increase usability for hybrid simulation experiments and make the tool generally more suitable for educational use.

4 Methodology

The research I have already completed has exposed many of the surface-level problems in developing an XML-based language to describe network simulation models and experiments. There exist other tools for XML manipulation which I have begun to investigate. On the document transformation end, these include data binding libraries like the Python XML data binding library, PyXB [8], and the more expressive schema definition languages such as RELAX NG [9] or Schematron [10]. These tools provide functionalities beyond those of the standard W3C tool set, such as XML marshalling and unmarshalling, and rules-based document validation.

I will investigate how other researchers have approached this problem and the strengths and weaknesses in their work. With a thorough analysis of related research literature and existing solutions, I will develop a thesis which is a meaningful and creative addition to the field. In the end, I will compare my work with the findings of other researchers in order to provide an objective analysis of its strengths and points that need improvement.

5 Conclusion

There is a great deal of research literature touting the benefits of XML as a simulation model description language and it is impossible to identify every way in which researchers have leveraged XML towards their particular goals. A selection of these advantages follows:

- XML-serialized model parameters make it easier to add new features, plugins, and third-party add-ons to a simulation framework [11];
- Interoperability among different simulator engines importing the same model description allows for cross-platform research [12];
- XML lends itself naturally to web-based client-server communication, allowing for easier distributed network simulation [12];
- XML allows for declarative specifications that ease database integration, user readability and the development of graphical user interfaces [6];
- The existence of XML-related technologies reduces development time and bolsters its future dependability [5].

It is agreed upon by virtually every researcher in the field that XML has the *potential* for providing much-needed standardization and portability to ad-hoc network simulation. Yet concrete implementations of XML-based model description have yet to be fully realized by the community of network simulation researchers at large. My thesis work will demonstrate what can be accomplished with the use of XML technologies for model construction, validation, and the generation of code that will execute in the *ns-3* simulator. Certainly *ns-3* could greatly benefit from such an initiative, and it is my intent to provide this.

References

[1] J. Liu, L. F. Perrone, D. M. Nicol, M. Liljenstam, C. Elliot, and D. Pearson, "Simulation modeling of large-scale ad-hoc sensor networks," in "Proceedings of the 2001 European Simulation Interoperability Workshop," (London, England, 2001).

[2] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies: The incredibles," Mobile Computing and Communications Review, **9 4**, 50 (2005).

[3] L. F. Perrone, C. Cicconetti, G. Stea, and B. C. Ward, "On the automations of computer network simulators," in "2nd International Conference on Simulation Tools and Techniques (SIMUTools)," (ICST and ACM, 2009).

[4] L. F. Perrone, C. J. Kenna, and B. C. Ward, "Enhancing the credibility of wireless network simulations with experiment automation," in "WIMOB '08: Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication," pp. 631–637 (IEEE Computer Society, Washington, DC, USA, 2008).

[5] F. Rioux, F. Bernier, and D. Laurendeau, "Design and implementation of an XML-based, technology-unified data pipeline for interactive simulation," in "Proceedings of the 2008 Winter Simulation Conference," pp. 1130–1138 (IEEE, 2008).

[6] M. Röhl and A. M. Uhrmacher, "Composing simulations from XML-specified model components," in "Proceedings of the 2006 Winter Simulation Conference," pp. 1083–1090 (IEEE, 2006).

[7] A. Hallagan, B. Ward, and L. F. Perrone, "An experiment automation framework for ns-3," in "SIMUTools '10: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques," pp. 1–2 (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2010).

[8] "PyXB Python XML Schema Bindings," Available at `http://pyxb.sourceforge.net/index.html` [Access September 6, 2010].

[9] "RELAX NG," Available `http://www.relaxng.org` [Access September 6, 2010].

[10] "ISO Schematron," Available at `http://www.schematron.com` [Access September 6, 2010].

[11] D. C. Chatfield, T. P. Harrison, and J. C. Hayya, "Xml-based supply chain simulation modeling," in "WSC '04: Proceedings of the 36th conference on Winter simulation," pp. 1485–1493 (Winter Simulation Conference, 2004).

[12] R. Canonico, D. Emma, and G. Ventre, "An XML description language for web-based network simulation," in "Proceedings of the Seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications," (IEEE, 2003).