

MODELADO DE PRESTACIONES DE SISTEMAS DE TIEMPO REAL MOPRESIT TIC2003-06293

Ramon Puigjaner*
Main Researcher

Carlos Juiz**
Researcher

Abstract

Designing real time systems needs to consider the non-functional requirements of temporal type, if possible since the first phases of the design. If not, the fulfilment of the requirements is just verified after the implementation. If they are not met, it is necessary to go back and reprogram and even redesign the system.

This project intends to develop a tool allowing the prediction, since the first phases of the design, of what will be the temporal behaviour of the system and, in this way, reduce the risk of erroneous decisions taken during the design. Obviously, for doing this prediction, it is necessary to complement the design information with other concerning the temporal behaviour of the system, which will be included in the system as annotations.

In order to experiment the best way to annotate designs, two notably different design approaches will be used: MASCOT and UML. The first one, originated for the design of hard real-time systems, is a method oriented to the data flow and based on activities interchanging information through intercommunication data areas. The second one, result of merging a set of diagrams proposed by several authors, allows the description of the different dimensions needed for the design of real-time systems of great size.

From the annotated designs in these two methods, we intend to that the tool derives queuing network models able to be processed by simulation methods and, if possible, by analytical methods. Till here this is the original abstract of the project. However, as it will be explained in the body of this presentation, several goals have been modified.

Keywords: Soft real-time systems. Performance modelling. Software performance engineering.

1. Project goals

1.1 Original goals

1. To build a tool allowing the automatic derivation of performance models based on queuing networks in order to know the performance of soft real-time systems before their implementation and their test verification of their timing requirements. Building this tool will be done first with a high level proprietary software, and then, with free software in order to ease its diffusion.
2. Building this tool should be preceded by the study of how to incorporate to the design languages, like UML, the timing aspects related with the system performance. To do this, it

* Email: putxi@uib.es

** Email: cjuiz@uib.es

will be necessary to see how to annotate UML and how can be extended the different UML diagrams in order to correctly construct the performance models and then to execute them.

3. Another approach to build the tool is derived from MASCOT as design method for real-time systems. This requires to complete the taxonomy of data interchange protocols to collect all the aspects related to the performance modelling and then to set up the tool allowing deriving queuing network models.

1.2 Modified goals

Due to several reasons (several members of the research team left the university, discovering new aspects interesting to be researched, etc.), the current goals of the project are:

1. The goal 3 has been worked out but instead of creating a tool for deriving performance models, the annotations of a MASCOT design have been standardised and a simulation process has been set up directly using the design description together with the annotations to obtain the desired performance estimations.
2. The goal 1 has been modified in order to develop PMIF, Performance Model Interchange Format, a mechanism whereby system model information may be transferred among performance modelling tools. Also it has been demonstrated how Web services can be used to facilitate the use of modelling tools that can interface with the PMIF. Indirectly, it also includes part of the goal 1 because one type of the performance models considered is the one based on UML mainly as input to obtain some type of performance model able to be computed to obtain performance results.
3. A new topic that has been included in the project goals is the performance assessment of ambient intelligence systems through ontology engineering. Thus, firstly it has been necessary to find appropriate description methods for distributed intelligent applications. Derived from the system characterization, typical software performance engineering techniques are based on the augmented description of the model regarding performance annotations. However, these annotations are only related with the syntactical view of the software architecture. In the next generation of performance assessment tools for ambient intelligent systems, the description of the system has to be able of reasoning and acquiring knowledge about performance. Having an appropriate architectural description including performance aspects, any possible design options for intelligent distributed applications can be evaluated according to their performance impact.

2. Main results related to each goal

2.1 MASCOT performance models (MASCOTime)

Soft real-time systems differ from traditional software systems in that they have a dual notion of correctness. That is, in addition to producing the correct output, soft real-time systems must produce it in a timely manner.

One of the most usual approaches to software engineering is based on the concept of a system as a set of interacting components. In order to characterize the interaction between the components is also necessary to express a protocol representing the timing constraints on the interaction and to its support by an explicit interconnection in the future implemented system.

MASCOT (Modular Approach to Software Construction Operation and Test) is a design and implementation method for real-time software development and brings together a co-ordinated set of techniques for dealing with the design, construction (system building), operation (run-time execution) and testing software, as its acronym wants to depict [1]. At the heart of MASCOT there

is a particular form of software structure supported by complementary diagrammatic and textual notations. These notations give visibility to the design as it emerges during development, and implementation takes place via special construction tools that operate directly on the design data. MADGE (Mascot Design Generator) is one of these tools [2]. Design visibility greatly eases the task of development management, and the constructional approach ensures conformity of implementation to design, and allows traceability of performance back to design. The functional model allows a system to be described in terms of independently functional units interacting through defined transfers of information. Here is introduced the notion of protocol component which is used to annotate the interaction between processes with a MASCOT representation [3]. Unfortunately, MASCOT does not provide any kind of performance constraints annotation on its functional components that could estimate, running a performance evaluation tool, whether the system being designed will meet or not the performance requirements.

In this research, we prototyped MASCOTime as the main contribution to the integration of performance engineering practices into a traditional system engineering methodology through performance constraint annotations. MASCOTime interferes minimally with the functional description on original MDL (MASCOT Description Language). In this way, system/software engineers do not perceive any change in the functional description of the future system.

However, once the system design method has been augmented with performance annotations, the information provided for the explicit system design is not evaluated itself. It is necessary either to build a performance tool to directly compute the performance of the model that MASCOTime generates. Therefore, we also build a prototype of discrete-event simulator for MASCOTime designs. The implementation of the simulator has been developed using objects that represent directly the MASCOTime components. Thus, the simulated components that build a system and the objects that the simulator manages are isomorphic. Due to the restrictions that would be imposed to the functional (and performance) model to apply analytical or numerical algorithms, we decide to avoid transforming the MASCOTime model to a traditional formal model for performance evaluation. The discrete-event simulator uses a direct translation between the MASCOTime components and their isomorphic Java objects. With this newly started work our goal is to automatize the analysis, that is, to build some expert knowledge into the simulator and make it analyze the system, diagnose it and give concrete advice to the system designer.

2.2 Performance Model Interchange Format (PMIF)

The Software Performance Engineering (SPE) process uses multiple performance assessment tools depending on the state of the software and the amount of performance data available. Web Services are distributed software components that allow the communication among applications or application components in a standard way through common protocols that are independent of the programming language, platform, presentation format, and operating system.

If a software performance modelling tool, like *SPE-ED* [4], wants to access another performance model Web Service, the tool first exports the model to a predefined model interchange format and then makes an HTTP call (application-to-application) to this Web Service. A tool owner who wants to provide a performance model Web Service needs to implement the Web Service, the import mechanism from the interchange format to a model in the tool's format, solve the model and send back to the client the obtained results. A user of several tools that support a common interchange format can create a model in one tool, and later move the model to other tools for further work without the need to laboriously translate from one tool's model representation to the other and the need to validate the resulting specification. Web Services provide a mechanism to easily invoke other tools automatically rather than having to manually execute each one.

We could also use performance model Web Services for SPE. Standard model interchange formats are the foundation that enables a Performance model Web Service. A common set of XML based interchange formats lets one use a variety of different tools as long as they support the format. To use an interchange format, each tool must either provide an explicit import and export command, or provide an interface to/from a file. With a file interface, an Extensible Stylesheet Language translation (XSLT) [5] can convert between the interchange format and the file.

There are two performance model interchange formats: the Performance Model Interchange Format (PMIF) is for the exchange of Queueing Network Models (QNM); and the Software Performance Model Interchange Format (S-PMIF) is for the exchange of software performance models among (UML-based) software design tools and software performance engineering tools. The PMIF meta-model is a model of the information that goes into constructing a QNM model. The PMIF was subsequently enhanced, implemented in XML, and named PMIF 2.0.

Thus, the PMIF 2.0 is a common representation for system performance model data that can be used to move models among system performance modeling tools that use a queueing network model paradigm. Earlier work also defined an SPE meta-model that formally defines the information required to perform an SPE study [6]. This model is known as the SPE meta-model because it is a model of the information that goes into constructing an SPE model. The S-PMIF based on the SPE meta-model is a common representation that can be used to exchange information between UML-based (Unified Modeling Language, [7]) software design tools and software performance engineering tools.

Our ongoing project intends to automate SPE performance evaluation. We have developed a Web Service prototype that makes implementation of a Performance model Web Service viable. Results would then be exchanged in the reverse direction and ultimately the software specialist would be able to view suggestions for performance improvements and automatically update the UML to reflect selected changes.

2.3 Performance Modelling of Ambient Intelligent Systems (PA-Ai)

During the last years UML has been widely used to specify, construct and document the functionality of software systems. In order to reduce the gap between functional models and performance formalisms, a software and performance community has emerged to provide (automatically) accessible techniques and tools to include performance annotations for building performance models of software models, constituting a new research topic known as Software and Performance Engineering (SPE). However, the growing availability of wireless networks and the expansion of powerful mobile devices define new challenges for these software distributed systems. Moreover, ambient intelligent applications designed for mobile computing are expected to run in a highly heterogeneous, dynamic and self-controlled environment. In that sense, less attention has been paid to these last phenomena in the SPE community because the traditional software architectures for distributed applications are difficult to translate to ad-hoc communication environments [8].

The conceptual framework of ambient intelligent systems includes an ontological foundation that structures the adaptability of the system to the users. An ontology is a formal, explicit specification of a shared conceptualization [9]. Conceptualization is referring to an abstract model of some piece of the real world, where the concepts that build the ontology are clearly identified. Therefore, the conceptualization process means the utilization of a formal language for representing the ambient. Clearly, this is a new challenge for the performance engineering area.

The importance of using an ontology as a conceptualization tool resides on the potential power of reasoning over the represented data. The ability of reasoning will allow taking decisions about performance and other non-functional features of systems. Different languages have been defined to represent particular knowledge, but none was totally compatible with the semantic World Wide Web as the Web Ontology Language (OWL). OWL is built on RDF and RDF schema [10] with a set of vocabularies for describing properties and classes. Since OWL includes semantics it is possible to apply logical rules to performance variables and even to infer new performance knowledge.

This research is fully complementary to previous approaches. For example, UML annotated diagrams that provide key information required for performance analysis describes, in different ways, behaviour of the system and consumption of resources, together. A huge number of approaches have been proposed to derive performance models from software architecture specifications: from annotated UML diagrams performance models are generated in the corresponding formalism (QN, SPN, SPA, etc.) and then they are off-line evaluated through analytical, numerical or discrete-event simulation techniques (see section 2.2). Very interesting examples of this R+D process can be found in the literature even the UML augmentation approach is not unique in the SPE, other software design methodologies has been extended to be simulated (see section 2.1).

Even the off-line performance evaluation relies on the syntactical annotated constraint values; it seems to be interesting to get information on-line about the relations in the context and to reason about them (monitoring and reasoning). The expressiveness of the ontology language rarely substitutes the syntactic richness of other approaches for an off-line performance evaluation of the system. However, the semantics and the standardization of OWL persuades to further research on this direction.

3. Results' indicators

Results appearing in some of these sections are due to the task of Catalina Lladó, in addition to those of the two authors of this report. The rest of members of the research group are mainly involved in the project TIC2001-1374-C03-01, reported below.

3.1 Graduate and Undergraduate Students

3.1.1 Katja Gilly de la Sierra

After obtaining her Research sufficiency, she is continuing her task, started in TIC2001-1374-C03-01, towards her PhD thesis. She is *Profesor Titular de Escuela Universitaria (i)* at the Universidad Miguel Hernández of Elx.

3.1.2 Salvador Alcaraz

After obtaining his Research sufficiency, he is continuing his task, started in TIC2001-1374-C03-01, to find the niche of his PhD thesis. He is *Profesor Titular de Escuela Universitaria* at the Universidad Miguel Hernández of Elx.

3.1.3 Carlos Guerrero

After finishing his doctoral courses, he is starting his task to find the niche of his PhD thesis. He was the last two year contracted as *Ayudante*, and from next academic year as *Profesor Colaborador* at UIB.

3.1.4 Jerònia Rosselló

She has been working, first as undergraduate student and then as graduate student in the domain of PMIF, from the beginning of the project until February 2005. She has obtained her degree of Informatics Engineer during her stay in the research group and has participated in several publications.

3.1.5 Pere Pau Sancho de la Jordana

He has been working, first as undergraduate student and after as graduate student in the domain of MASCOTime, since the beginning of the project. Initially he was technician contracted by the project and then with a similar status but partially funded also by the UIB. He has obtained his degree of Informatics Engineer during his stay in the research group; currently he is working in his PhD Thesis and has participated in several publications.

3.1.6 Isaac Lera

He is working as undergraduate student in the domain of performance of Ambient Intelligent systems, since July 1st. He is contracted as technician by the project. He has participated in several publications.

3.2 Publications

- 1 Haring, G.; C. Juiz, C. Kurz, R. Puigjaner, J. Zottl, *Framework for the Performance Assessment of Architectural Options on Intelligent Distributed Applications*. PERMIS'04. Proceedings. Multimedia publication.
- 2 Molero, X., C. Juiz, M.J. Rodeño, *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos*. Prentice-Hall. Book ISBN 84-205-4093-5.
- 3 Sancho, P.P., C. Juiz, C., R. Puigjaner, *Evaluating the Performance of Architectures in MASCOTime*. SERP'04. Proceedings CSREA Press Book Chapter ISBN 1-932415-28-9.
- 4 Sancho, P.P., C. Juiz, C., R. Puigjaner, *Integrating System Performance Engineering into MASCOTime Methodology through Discrete-event Simulation*. In *Applying Formal Methods: Testing, Performance, and M/E-Commerce* (N. Núñez, Z. Maamar, F. L. Pelayo, K. Pouttschi, F. Rubio, eds.). Proceedings of EPEW 2004. Book Chapter ISBN 3-540-23169-2. Journal article: LNCS 3236, ISSN 0302-9743. pp. 278-292.
- 5 Smith, C. U. and C. M. Lladó, *Performance Model Interchange Format (PMIF 2.0): XML Definition and Implementation*, Proc. 1st Int. Conf. on Quantitative Evaluation of Systems (QUEST), Enschede, NL, IEEE Computer Society, 2004. Book Chapter ISBN: 0-7695-2185-1
- 6 Juiz C., J. Zottl, G. Haring, R. Puigjaner, *Performance Assessment of Architectural Options on Intelligent Distributed Systems*. IWSSA'05. Proceedings. Book Chapter. CSREA Press, vol I, 2005. ISBN 1-932415-49-1. pp. 45-51.
- 7 Kurz, C., C. Guerrero, R. Puigjaner, G. Haring, *Extending TPC-W to Allow for Fine Grained Workload Specification*. Proceedings of the 5th International Workshop on Software and Performance, WOSP 2005. Book Chapter. ACM Press, 2005. ISBN 1-59593-087-6. pp. 167-174.
- 8 Lera, I., C. Juiz, R. Puigjaner, *Web Operational Analysis Through Performance-Related Ontologies in OWL for Intelligent Applications*. Web Engineering: 5th International Conference, ICWE 2005 (D. Lowe, M. Gaedke, editors). LNCS 3759. Springer S+B M 2005. Book Chapter. ISBN 3-540-27996-2. Journal article: ISSN 0302-9743

- 9 Lera, I., C. Juiz, C. Kurz, R. Puigjaner, G. Haring, J. Zottl, *Performance Assessment on Ambient Intelligent Applications through Ontologies*. Proceedings of the 5th International Workshop on software and Performance, WOSP 2005. Book Chapter. ACM Press, 2005. ISBN 1-59593-087-6. pp. 195-204.
- 10 Rosselló, J., C. M. Lladó, R. Puigjaner, C. Smith, *A Web Service for solving Queuing Network Models using PMIF*. Proceedings of the 5th International Workshop on software and Performance, WOSP 2005. Book Chapter. ACM Press, 2005. ISBN 1-59593-087-6. pp. 187-192.
- 11 Rosselló, J., C. M. Lladó, R. Puigjaner, C. Smith, *A Lexical and Syntactical Analyzer for the Exporting of QNAP to the Performance Model Interchange Format (PMIF)*. 31 Conferencia Latinoamericana de Informática, CLEI 2005. Accepted paper.
- 12 Sancho, P.P., C. Juiz, R. Puigjaner, *Evaluating the Performance of Architectures in MASCOT*. Science of Computer Programming, Volume: 57-1, Journal article ISSN 0167-6423. pp. 45-60.
- 13 Sancho, P.P., C. Juiz, R. Puigjaner, *Giving Feedback on MASCOTime Simulation Results*. SERP'05 Proceedings. Book Chapter. CSREA Press, vol. I, 2005. ISBN 1-932415-49-1. pp. 126-134.
- 14 Sancho, P.P., C. Juiz, R. Puigjaner, *Automatic Performance Evaluation and Feedback for MASCOT Designs*. Proceedings of the 5th International Workshop on software and Performance, WOSP 2005. Book Chapter. ACM Press, 2005. ISBN 1-59593-087-6. pp. 193-194.
- 15 Smith, C. U., C. M. Lladó, V. Cortellessa, A. Di Marco and L. G. Williams, *From UML Models to Software Performance Results: An SPE Process Based on XML Interchange Formats*, Workshop on Software and Performance, WOSP 2005. Book Chapter. ACM Press, 2005. ISBN: 1-59593-087-6

3.3 Joint Actions and Projects

3.3.1 Integrated Action with the University of Vienna (Austria)

This project aims to develop a methodology framework for performance engineering of applications with ambient intelligence, showing its applicability in a few test cases. As a result of the encouraging initial results obtained in this integrated action, the goals of the project have been redefined.

3.3.2 *Análisis de las prestaciones en los entornos web*

Coordinated Project (TIC2001-1374-C03-01) with the Universities *Politécnica de València* and *Oviedo*, initially planned to finished in December 2004 but prorogued until December 2005.

3.4 National and International Collaborations

Members of the group maintain regular collaboration actions with the following universities: Miguel Hernández, *Politécnica de València*, dell'Aquila (IT), Viena (AT), Paul Sabatier de Toulouse (FR), Imperial College (UK), Carleton (CA), etc.

3.5 Relevant Participation on Meetings

- 1 Galmés, S., C. Juiz, Lladó, C., Puigjaner, R., XXX Conferencia Latinoamericana de Informática. PC Members.
- 2 Galmés, S., C. Juiz, International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2004. PC Members.
- 3 Juiz C., Workshop on Software and Performance, WOSP 2005. Program Chair, Steering Committee Member and Session Chair.
- 4 Juiz, C., Workshop on System/Software Architectures IWSSA'05. PC Member.
- 5 Juiz, C., Conference on Software Engineering Research and Practice, SERP'05. PC Member.

- 6 Juiz, C., R. Puigjaner, International Symposium on Computer and Information Sciences, ISCIS 2004. PC Members.
- 7 Juiz, C., 8th WSEAS International Conference on COMPUTERS. Guest Editor and Session Chair.
- 8 Juiz, C., 12th Conference of the IEEE / ACM MASCOTS 2004. Session Chair.
- 9 Juiz, C., R. Puigjaner, Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2005. PC Members.
- 10 Lladó, C. M., R. Puigjaner, 31 Conferencia Latinoamericana de Informática, CLEI 2005. PC Members.
- 11 Puigjaner, R., 12th IEEE / ACM International Symposium on Performance Evaluation of Computer and Telecommunication Systems, MASCOTS 2004. PC Member.
- 12 Puigjaner, R., Network Control and Engineering for QoS, Security and Mobility, Net-Con 2004. Conference Chair.
- 13 Puigjaner, R., XXX Conferencia Latinoamericana de Informática CLEI 2004. Session Chair.
- 14 Puigjaner, R., IFIP World Computer Congress, WCC 2004. PC Member.
- 15 Puigjaner, R., 1st International Conference on Quantitative Evaluation of Systems, QEST 2004. PC Member.
- 16 Puigjaner, R., 3rd International Conference on Networking, NETWORKING'2004. PC Member and Session Chair.
- 17 Puigjaner, R., XXX Conferencia Latinoamericana de Informática, CLEI 2004. Session Chair.
- 18 Puigjaner, R., EPEW'2004. PC Member.
- 19 Puigjaner, R., Workshop on Software and Performance, WOSP 2005. General Chair, Steering Committee Member and Session chair.
- 20 Puigjaner, R., 4th International Conference on Networking, NETWORKING 2005. PC Chair and Session Chair.
- 21 Puigjaner, R., EPEW'2005. PC Member.

4. References

- [1] Simpson, H.R.: The Mascot method, Software Engineering Journal, pp. 103-120, May 1986.
- [2] Harding, M.D.: MASCOT DESIGN GENERATOR (MADGE) v.6.0. Release Notification, n. 5096, issue 11, November 1998.
- [3] Simpson, H.R.: Protocols for Process Interaction. 1, 2, 3. Matra BAe Dynamics, 2000.
- [4] Smith, C.U., and Williams L.G., Performance Engineering Evaluation of Object Oriented Systems with SPE•ED, in Computer Performance Evaluation: Modelling Techniques and Tools, Springer-Verlag, 1997.
- [5] Object Management Group (OMG): XML Metadata Interchange (XMI) Specification, version 1.2, <http://www.omg.org/cgi-bin/>, 2001.
- [6] Smith, C.U., and Williams L.G., Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software, Addison-Wesley, 2002.
- [7] Object Management Group (OMG): Unified Modeling Language Specification, version 1.3, <http://www.omg.org/uml>.
- [8] Beer, W., Christian, V., Ferscha, A. and Mehrmann L., Modeling Context-Aware Behavior by Interpreted ECA rules. In Proceedings of Euro-Par 2003, H. Kosch, L. Böszörményi and H. Hellwagner (Eds.), LNCS 2790, pp. 1064-1073, 2003.
- [9] Berners-Lee, T., Hendler, J. and Lassila, O. "The Semantic Web", Scientific American, May 2001.
- [10] OWL Web Ontology Language Overview <http://www.w3.org/TR/owl-features/>.