www.worldses.org

Proceedings of the 5th WSES International Conference o Circuits, Systems, Communications and Computers (CSCC 2001)

Technical Co-Sponsored by IEEE SP Society
ISBN: 960-8052-33-5

Proceedings of the 3rd WSES International Conference o Mathematics and Computers in Physics (MCP 2001)

ISBN: 960-8052-34-3

Proceedings of the 3rd WSES International Conference o Mathematics and Computers in Mechanical Engineerin (MCME 2001)

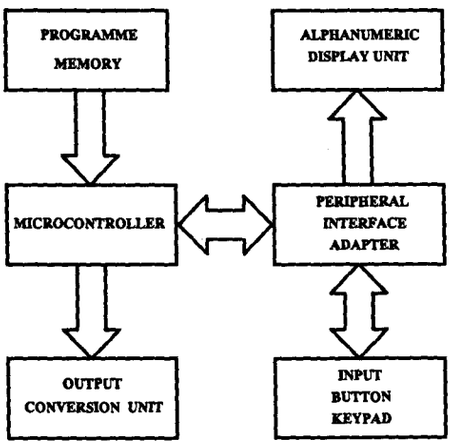
ISBN: 960-8052-35-1

Rethymno, Greece,
July 8-15, 2001

- CSCC '01 - MCP '01 - MCME '01 Papers
- Edited by Nikos E. Mastorakis (mastor@ieee.org)
- Install Acrobat Reader

Electrical and Computer Engineering Series
A Series of Reference Books and Textbooks

Advances in Signal Processing and Computer Technologies



G.Antoniou, N. Mastorakis, O.Panfilov
Editors

WSES http://www.worldses.org

Electrical and Computer Engineering Series
A Series of Reference Books and Textbooks

Advances in Signal Processing and Computer Technologies

http://www.worldses.org

Edited by

G.Antoniou
Montclair State,
University Montclair,
NJ 07043, USA

N.Mastorakis
Depart.of Computer Science,
Milit. Inst. Univ.Education,
Hellenic Naval Academy,
Hatzikyriakou, 18539,
Piraeus, GREECE.

O.Panfilov
Motorola
1501 West Shure Drive,
Arlington Heights, IL 60004,
USA

Advances in Signal Processing and Computer Technologies

Published by WSES Press
http://www.worldses.org

ISBN: 960-8052-37-8

http://www.worldses.org

Preface

This book contains significant results in Signal Processing and various Computer Technologies: It contains the latest developments and reflects the experience of many researchers working in different environments (universities, research centers or even industries), publishing new theories and solving new problems in various branches of science and engineering. The book consists of two parts. The first part is dedicated to signal processing while the second part to computer (software and hardware technologies.

The main objective of the book is the interconnection of diverse scientific fields of mathematical science, the cultivation of every possible scientific collaboration, the exchange of views and the promotion of new research targets as well as the further dissemination, the dispersion, the diffusion of the scientific knowledge.

We are sure that the efforts of the authors to provide high level contributions will be appreciated by the relevant scientific and engineering community.

We are convinced that the book will be a source of knowledge and inspiration for all academic members, researchers and practitioners working on the topics covered by the book.

We cordially thank all the people of WSES and WSES Press for their efforts to maintain the high scientific level of conferences, proceedings and post-conference books.

We feel very much indebted to them for their invaluable help to keep this level at the high standard that our scientific community has grown accustomed to expect from it.

The Editors

A Tool for the Management of the Software Maintenance Process

FELIX GARCIA, LUIS MARQUEZ, FRANCISCO RUIZ, MARIO PIATTINI, MACARIO POLO
Alarcos Research Group
Escuela Superior de Informática, University of Castilla-La Mancha
Ronda de Calatrava, s/n. 13071, Ciudad Real
SPAIN
{fgarcia |lmarquez}@proyectos.inf-cr.uclm.es; fruiz@inf-cr.uclm.es

Abstract: Software Process Management is a complex process. Due to the large number of different aspects that have to be considered it is useful to establish a conceptual architecture. Owing to the definition of conceptual levels in this architecture it is possible to work more easily with different models and metamodels of software processes.

In this paper we present MANTIS-Metamod, a tool for representing and managing software process metamodels based on the MOF (Meta-Object Facility) standard proposed by the Object Management Group [5]. This standard defines a framework using the principles of object oriented modelling for the definition of information models for metadata.

MANTIS-Metamod is a component of MANTIS, an integrated environment for the management of the Software Maintenance Process (SMP). MANTIS uses 4 conceptual levels (based on the architecture proposed by MOF) for dealing with the large quantity of different aspects that must be taken into consideration when modelling, managing, controlling and executing a software process. Although MANTIS is dedicated to SMP, the tool allows the meta-modelling of any software process.

Key words: Software Engineering Environment Tools, Software Process Metamodelling, MOF, XMI.

1 Introduction

Many studies have demonstrated that the majority of the overall expenses incurred by a software product throughout its lifecycle occur during the maintenance stage [7] and that the characteristics specific to this stage that differentiate it clearly from the development stage, make it very useful to have specific methods, techniques and tools at one's disposal [6].

Consequently it is necessary to define and construct an environment for the integral management of the software maintenance process, given the complexity of this type of projects (due to the size and complexity of the product to be maintained and the difficulty of the task to be undertaken).

The MANTIS project aims to define and construct an integrated environment for the management for the SMP ("big-E environment"). By using the nomenclature "big-E environment" our intention is to emphasize the idea that MANTIS is broader than the concepts of:

- Methodology (in its usual sense), that is to say, a series of related methods and techniques, and
- Software Engineering Environment (SEE), that is to say, a collection of software tools used to support software engineering activities [2].

MANTIS includes the different aspects that must be taken into account when undertaking software maintenance projects. For the integrated management of SMP, MANTIS integrates, amongst others, the people (with certain skills and they carry out certain roles in the project), the methodologies(that people use), the tools (to facilitate conforming with the standards) and the activities (in which the teams work and help to reach a milestone which indicates the progress of the process).

2 MANTIS and MOF.

An important principle of modern software engineering is the separation of a system in encapsulation layers which can mostly be specified, designed and constructed independently. Following this philosophy, in MANTIS we have defined 4 conceptual levels that are based on the MOF standard for object oriented modelling proposed by the Object

Management Group, [5]. In table 1 we can see these 4 levels of the MOF architecture and its adaptation to MANTIS.

Level	MOF	MANTIS
M3	MOF-model (Meta-metamodel)	MOF-model of SMP
M2	Meta-model	SMP metamodel
M1	Model	MANTEMA & others techniques (SMP concrete model)
M0	Data	Instances of SMP (real-world concrete software maintenance projects)

Table 1. Conceptual levels in MOF & MANTIS.

Examples of real and specific software maintenance projects with time and cost restrictions are found in the level M0. The data handled at this level are instances of the concepts defined at the higher level M1. The specific model that we use at level M1 is based on the MANTEMA methodology and a group of techniques adapted to the special characteristics of maintenance: effort estimation, risk estimation, process auditing [9] etc. Level M2 corresponds to the SMP metamodel. For example, the generic concept of Activity used in M2 is present in the activities "Problem Report Analysis" or "Modification Implementation" in M1 and these in turn appear in level M0 as "problem report analysis number 36 of the PATON project".

In the last conceptual level of MANTIS, M3, The SMP metamodel is represented in a MOF-model. A MOF-model is composed basically of two types of objects: MOF-class and MOF-association.

Consequently, all the concepts represented in level M2 are now considered instances of MOF-class and MOF-association. For example, Activity, Actor or Artefact will be instances of MOF-class ; and "Activity use Resource" or "Artefact is input of Activity" are instances of MOF-association. An MOF-model can be represented by UML diagrams or by MODL language (Meta Object Definition Language) but in order for it to be used automatically and to be portable amongst tools in a SEE, which is what interests MANTIS, it is much better to represent it by using one of the metadata interchange standards. For this reason, in MANTIS we use XMI (XML Metadata Interchange) [4], based on XML (eXtensible Markup Language), for storing the metamodels.

MOF is a model-driven distributed object framework for specifying, constructing, managing, interchanging and integrating metadata in software systems, thus enabling the flexible integration of systems. MOF describes an abstract modelling language aligned with the core of the UML of the OMG.

In the following diagram a summary of the class hierarchy that makes up the MOF-model is represented by means of an UML diagram.

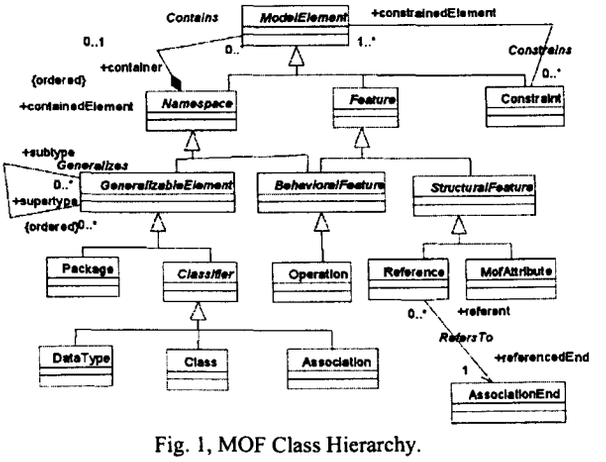


Fig. 1, MOF Class Hierarchy.

By incorporating this last conceptual level and using standards for metamodelling (MOF) and for metadata interchange (XMI) we can achieve as flexible as possible environment for defining and sharing models and metamodels. The inclusion of level M3 allows us to work with different versions of SMP metamodels, which is a requirement in order to be able to manage the process improvements.

3 Description of the Tool.

MANTIS-Metamod is a tool for representing metamodels of software processes based on the MOF standard proposed by the Object Management Group [5].

The tool provides support for the modelling of software processes in general (described according to the architectural levels of MOF). In our project (within the framework of the MANTIS project), however, it is applied for the modelling of the SMP.

As a result the tool in question constitutes the component of MANTIS that is responsible for the modelling of software maintenance processes.

In this paper we are going to describe the component of the tool responsible for the definition of the correspondences between the level 3 of the MOF model (based on MOF-classes, MOF-associations) and the level 2 (instances of the level 3).

With this component of the tool we aim to facilitate the integral management of the SMP by defining the metamodels needed for this management for which we will use a common terminology and the abstraction mechanism that provides level 3 of the architecture of MOF models as a base.

All the applications have a presentation code, a data processing code and a data storage code ("3 Tier Architecture") [3]. The architecture of an application is the conceptual view of its structure. Consequently, the architecture of the tool is composed of 3 components or vertical layers with the aim of reducing complexity and providing a degree of encapsulation. As we have already said, these levels are: interface, processing and storage.

The interface layer will be responsible for obtaining information from the user, sending the information from the user for processing and presenting the results to the user. The functions of the processing layer are to receive information entering at the presentation level , interact with the storage services in order to execute the operations, and send the processed result to the presentation level. Lastly, the storage layer is responsible for storing, retrieving and maintaining the data.

To receive user data, the application is composed of a metamodel administrator as its principal component and of a system of windows that allows a visual description of the classes that make up the core of the MOF model (Package, Class, Datatype, Attribute, Operation, Reference, AssociationEnd and Constraint).

The metamodel administrator has a three-shape structure, as does the MOF model: a package contains classes and associations, a class contains attributes and operations, an association contains restrictions etc…

The system of associated windows allows the description of the classes that make up the core of the MOF model. In figure 2 we can see the window associated to the definition of a MOF class.

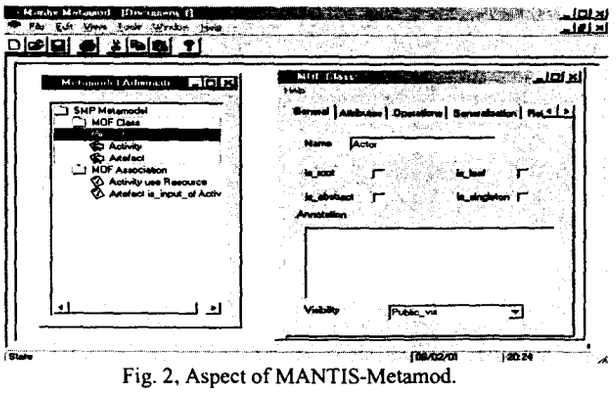


Fig. 2, Aspect of MANTIS-Metamod.

For example, an instance of MOF-class could be the level M2 class "Actor" belonging to the metamodel SMP. This instance of MOF-class, as can be seen in the diagram, makes it possible for any element of the model to use it (visibility=public_vis), it is not abstract (is-abstract=false), it can have supertypes (is-Root=false), it can be a supertype (is-leaf=false) and only one level M1 instance can exist at most (is-singleton=false).

The models defined for the user by means of the interface, are validated and internally represented according to the hierarchy of the classes described by the MOF model, specified in the IDL document attacked to the MOF standard.

For the storage of the metamodels (level 2 defined with MOF) we must use a representation that facilitates the interchange of information and as a result gives the tool great flexibility. With this objective in mind we will use XMI.

The main objective of XMI is to allow easy interchange of metadata between modelization tools (based on UML) and metadata tools and repositories (based on MOF) in distributed heterogenic environments.

XMI uses XML as syntax for interchange transference and format.

Therefore for the storage of metamodels (level 2) we must define a DTD (Document Type Definition) that represents MOF (level 3), and in this way all the stored XML documents must respect the rules defined in the DTD in order for them to be considered as valid instances of the MOF model.

The use of XMI facilitates and permits the interchange of metamodels based on MOF, so the tool must provide two fundamental services:

- The storage of the MOF models defined in the tool, in a local repository of metadata represented in XMI(for exportation).
- Importation of metamodels.

The storage layer provides the necessary services for the storage and extraction of metadata, on a metadata repository composed of XML documents.

The services are similar to those that a Management System for Relational Databases could provide with the peculiarity that in our case what we have at our disposal is a repository of metadata stored in XML documents, that is to say, we have a collection of structured documents.

For the storage of the MOF models in XML documents and for the extraction of information of these, a library of services is used, which in turn uses the services that the XML DOM (Document Object Model) provides. This model provides a collection of classes that represents the hierarchical structure on which all XML documents are based. The model provides classes that represent Documents, nodes, lists of nodes etc, with the properties and methods necessary for constructing XML documents using the hierarchical structure that characterizes them as a base.

The following is a scheme of the tool based on its storage layer and which sums up the previously mentioned concepts.

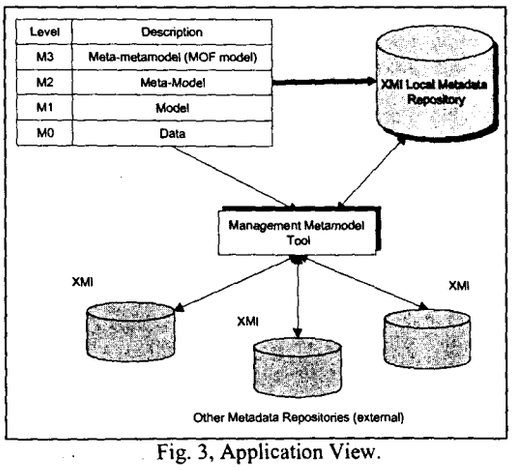


Fig. 3, Application View.

As shown in figure 3 the tool maintains and manages a local repository of metadata for the storage of the MOF models. Basically this diagram represents the two services provided by the storage layer:

- Storage of the MOF models defined by means of our tool in XMI format in order to facilitate their exportation.
- Importation of process models(defined according to MOF levels), using the interchange XMI format as a base in such a way that useable MOF models can be taken advantage of for the representation of software processes independent of the platform and the tool with which they have been defined.

To complement the tool proposed, other tools of visual modelling based on UML can be used for the visualization of the MOF models as class diagrams (eg. Rational Rose). This functionality is possible owing to the fact, as already mentioned, that the tool has the capacity for interchanging MOF models by means of the interchange of metamodels with XML standard (XMI).

4 Conclusions

In order to manage any software process -such as the PMS – in an integrated and structured way, it is useful to consider different levels of abstraction, both for modelling the process and managing the different instances of execution (specific projects).

In this paper we have presented MANTIS-Metamod, a tool for the description, importation and exportation of software process metamodels. With this component of MANTIS our aim is to take advantage of the proven benefits that can be obtained by using metamodels for the integral maintenance of software processes.

In MANTIS this integration is based on the use of a common terminology and on the abstraction mechanism provided by the architecture of conceptual levels described by the MOF standard.

A fundamental use of the tool is that it constitutes the necessary support for the improvement of the software process. In the context in which the tool is set, owing to the definition of appropriate projections between the MOF levels 2 and 3, it is possible to incorporate the subontology of measure to the SMP metamodel, by adding the necessary classes and associations.

Currently we are working on increasing the functions of our tool. One of the main short-term objectives is to develop components that would be responsible for providing support to the projection between the levels M2-M1 and M1-M0, in such a way that all the concepts dealt with in software process development can be adequately integrated and handled (from modelling to the management of the execution of specific projects).

Acknowledgements: This work has been undertaken in collaboration with the company Atos ODS subsidized by the MANTIS and MPM projects. MANTIS has been partially supported by the European Union and CICYT-Spain(1FD97-1608TIC). MPM has been partially supported by the Science and Technology Ministry of Spain(FIT-070000-2000-307).

References:

[1] Becker-K., U., and Webby, R., A Comprehensive Schema Integrating Software Process Modelling and Software Measurement, Fraunhofer Institute, *IESE report N° 047.99/E.*, v. 1.2, 1999.

[2] *ISO/IEC JTC1/SC7/WG4 15940 working draft 5*: Information Technology -Software Engineering Environment Services, june-2000.

[3] Larman, C. *Applying UML and Patterns*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[4] OMG XML Metadata Interchange (XMI), v. 1.3, March-2000.

[5] OMG Meta Object Facility (MOF) Specification, v. 1.3, March 2000. In http://www.omg.org.

[6] Piattini, M.; Ruiz, F.; Polo, M., Bastanchury, T., Fernández, I., Martínez, M.A. *Mantenimiento del Software: Conceptos, Métodos, Herramientas y Outsourcing*. Ed Ra-Ma, Spain 1988.

[7] Pigoski, T.M., *Practical Software Maintenance. Best Practices for Managing your Investment*. Ed. John Wiley & Sons, USA 1996.

[8] Polo, M., Piattini, M., Ruiz, F., Calero, C., MANTEMA versión 2.0: una Metodología para el Mantenimiento del Software. *UCLM, Dep. de Informática, informe técnico UCLM-DI-99-01.*

[9] Ruiz, F., Piattini, M., Polo, M., Calero, C., Audit of Software Maintenance. *In "Auditing Information Systems"*. Idea Group Publishing, USA 2000.

[10] WfMC TC00-1003 1.1: Workflow Management Coalition. The WorkflowReference Model, june-1995.