PROCEEDINGS OF THE INTERNATIONAL SYMPOSIUM ON INFORMATION SYSTEMS AND ENGINEERING

ISE'2001

Editors:

Waleed W. Smari, Nordine Melab, Kokou Yetongnon

Associate Editors:

Ulrich Frank, Nazli Goharian

Las Vegas, Nevada, USA

June 25 - 28, 2001

ISBN: 1-892512-85-8

Printed in the U.S.A.

Foreword

Welcome to the First International Symposium on Information Systems and Engineering. This symposium provides a forum to address, explore and exchange information on the state-of-the-art in information technology, engineering, and systems, their design and use, and their impact. It's intent is to bring together scientists, engineers, practitioners, computer users, and students to share and exchange their experiences, ideas, and research results about all aspects of Information Systems and Engineering; to discuss challenges encountered; and to seek potential solutions. Hopefully, the symposium will provide opportunities to build new bridges and develop future collaborations.

The Information Systems and Engineering field has been intensively developing during recent years with the emergence of new technologies. Many experimental as well as commercial systems continue to be built at all levels, including hardware, software algorithms and applications that integrate these new technologies. The design and development of such systems require the researcher to tackle a wide range of issues and to be familiar with newly introduced techniques and products. The papers published in these proceedings represent results from researchers in various Information Systems and Engineering (ISE) areas.

We would like to thank all authors who submitted their work to ISE 2001 and who are presenting their papers in Las Vegas. If the papers and presentations are the essence of what makes ISE possible, then the International Program Committee is the key to its successfulness. The IPC members and reviewers did an exceptional job and we are grateful for their helpful efforts in reviewing and evaluating the submissions.

We also wish to thank Dr. Ulrich Frank and his research team at the Institut für Wiertschaftsinformatik, Universitat Koblenz-Landau for the organization of the *Workshop on Knowledge Management Systems: Concepts, Technologies and Applications.* We thank Ms. Nazli Goharian for having organized the *Special Session on Large Scale Information Processing.*

We must also acknowledge here Dr. Iyad Ajwa, our Publicity Chair for his contributions and support. There are many people in the organizing committees of the Multi-conferences and the co-sponsors who must also be recognized for their hard work, assistance and dedication, without which this symposium would not have been possible. Finally, we are thankful to Professor Hamid Arabnia, the General Chair of the 2001 Multi-conferences, for kindly including the ISE'2001 Symposium as part of the Multi-conferences this year. We thank him also for the constant and continual cooperation and assistance we received throughout the year.

We thank you, our attendees, for making ISE 2001 possible. Enjoy the symposium and consider including this symposium in next year's plans.

Waleed W. Smari, Nordine Melab and Kokou Yetongnon
Symposium Co-Chairs, ISE 2001

ISE 2001 Program

We are pleased to present the program for the first ISE symposium. This program includes a total of 86 contributions distributed as follows: 60 regular research papers (RRP), 24 regular research reports (RRR), and 2 short research papers (SRP).

The main track of the symposium consists of 35 RRP manuscripts. Each of these received a minimum of two rigorous reviews plus three independent evaluations. Acceptance rate was 62%.

The *Workshop on Knowledge Management Systems: Concepts, Technologies and Applications* screened the submissions to a total of 11 RRP manuscripts, with the review process being handled by the workshop organizers. A similar count occurred for the manuscripts that came from the *Special Session on Large Scale Information Processing*. The paper reviews were handled by the session organizer. In addition, a total of 10 RRR contributions came from the Korean Information Processing Society (KIPS), while other sponsors and affiliated organizations (CSREA, IPSJ - SIGMPS, ITI, JHPC, and WSES) contributed 3 RRP, 14 RRR and 2 SRP.

The technical papers in the program represent works from academia, research laboratories, government, and industry. The submissions came from 21 countries, with the top five being USA (32%), Korea (16%), France (13%), China (10%), and Germany (10%).

The four-day program consists of an Opening Remarks Session, a Keynote Speech, 6 Discussion Sessions, 16 formal Technical Sessions, a Panel Discussion Session, 12 short breaks, and a Reception.

On behalf of the organizers, the IPC and all those who made ISE 2001 possible, I hope you find this volume interesting, thought provoking and beneficial.

Thank you.

Waleed W. Smari
International Program Committee

International Program Committee - ISE 2001

Akhter, F., Zayed University, United Arab Emirates
Alkhatib, G., King Fahd University of Petroleum and Minerals, Saudi Arabia
Alpaslan, F.N., Middle East Technical University, Turkey

Benslimane, D., Université de Bourgogne, France
Benatallah, Boualem, University of New South Wales, Australia
Boulmakoul, A., Universite Mohamadia, Morrocco
Bouneffa, M., Université du Littoral-Côte d'Opale, France
Bouzeghoub, M., Université de Versailles, France

Chen, Shu-Ching, Florida International University, USA
Claramunt, C., University of Nottingham, United Kingdom
Daoudi, M., Université de Oujda, Morrocco
Djeraba, C., Université de Nantes, France

Geib, J.M., Université de LilleI, France
Guizani, M., University of West Florida, USA
Kabalan, K.Y., American University of Beirut, Lebanon
Koukam, A., Université de Technologie de Belfort-Montbeliard, France

Laskri, M.T., Université d'Annaba, Algeria
Lee, T.K., University of Singapore Kent Ridge - Singapore
Maamar, Z., Zayed University, United Arab Emirates
Marciano, J.P., Université Aix-Marseille, France
Mendes, E., University of Auckland, New Zealand
Monties, S., Ecole Polytechnique Fédérale, Switzerland

Patel, D., South Bank University, United Kingdom
Rolland, C., Université de Paris I, France
Samaras, G., University of Cyprus, Cyprus

Rana, O., University of Wales-Cardiff, United Kingdom
Tari, Z., Royal Melbourne Institute of Technology, Australia
Trehel, M., Université de Franche-Comte, France
Zimanyi, E., Université Libre de Bruxelles, Belgium
Zubairi, J.A., SUNY at Fredonia, USA

SESSION: *Software Specification, Framework, and Development*

SESSION: *Large Scale Information Processing*

Managing complexity of software processes

Luis Marquez, Felix Garcia, 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
Phone +34926295300 (ext 3747)
SPAIN
lmarquez @proyectos.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. The use of Modeling and Metadata enabled architectures allows us to manage the complexity of software process.*

In this paper we present a 4 level conceptual architecture for representing and managing software process metamodels based on the MOF (Meta-Object Facility) standard proposed by the Object Management Group (OMG, 2000). In particular, we'll apply these concepts to the maintenance software process.

As support we present MANTIS-Metamod, a tool of metamodeling software processes based on the concepts previously commented. 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). Although MANTIS is dedicated to SMP, the tool allows the meta-modeling of any software process.

Key words: Software Process Metamodeling, MOF, XMI.

1 Introduction

A process is a set of interrelationated activities that converts inputs in outputs [2]. A software process consists in a set of concurrent and cooperative activities, that are related with the development and maintenance of software, so as with the project management and product quality [7].

Software processes are inherently complex. They involve many people, each of them with individual responsabilities and skills, and they produce or modify a whole range of intermediate or deliverable artifacts [1]. Due to this complexity it is convenient to implement a conceptual architecture, using an important principle of modern software engineering: the separation of a system in encapsulation layers which can mostly be specified, designed and constructed independently.

Typically 4 layers are considered adequate. The "meta" prefix indicates one level of abstraction higher than root and it is used in a relative manner (a metamodel is a model of a model, a meta-metamodel is a model of a metamodel and so on).

Metamodel help us understand and describe the problem domain, help others understand the problem domain by using the same language, define a vocabulary for the elements in the problem domain and manage complexity by raising the level of abstraction at which we think and design. Metadata enabled Systems are used to manage complexity of systems and applications.

On other hand, Modeling and Metadata standards are necessary for interchange of software artifacts and interoperability between tools, applications, middleware and data stores.

And so on, in order to manage the software process complexity, it's necessary to establish an open conceptual level architecture providing the encapsulation needed and allowing to work easily with a great amount of data and metadata.

2 Software Process Metamodeling

As example of software process we are going to consider the maintenance software process.

Many studies have demonstrated that the majority of the overall expenses incurred by a software product throughout its lifecycle occur during the maintenance stage [9] and that the characteristics specific to this stage that differentiate it clearly from the development

...age, make it very useful to have specific methods, techniques and tools at one's disposal [8].

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" the 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 [3].

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).

Following this philosophy, in MANTIS we have defined 4 conceptual levels that are based on the MOF standard for object oriented modeling proposed by the Object Management Group, [6]. 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 [11] etc. Level M2 corresponds to the SMP metamodel. For example, the generic concept of Maintenance Activity used in M2 is present in the activities "Problem Report Analysis" or "Modification Implementation" or "Corrective Maintenance Activity" in M1 and these in turn appear in level M0 as "Corrective Maintenance Activity number 36 of the PATON project of the MAINTENANCE SOLUTIONS S.A enterprise".

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 Maintenance Activity 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 Software Engineering Environment, 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) [5], based on XML (eXtensible Markup Language), for storing the metamodels.

In the following figure we can observe the example commented before:

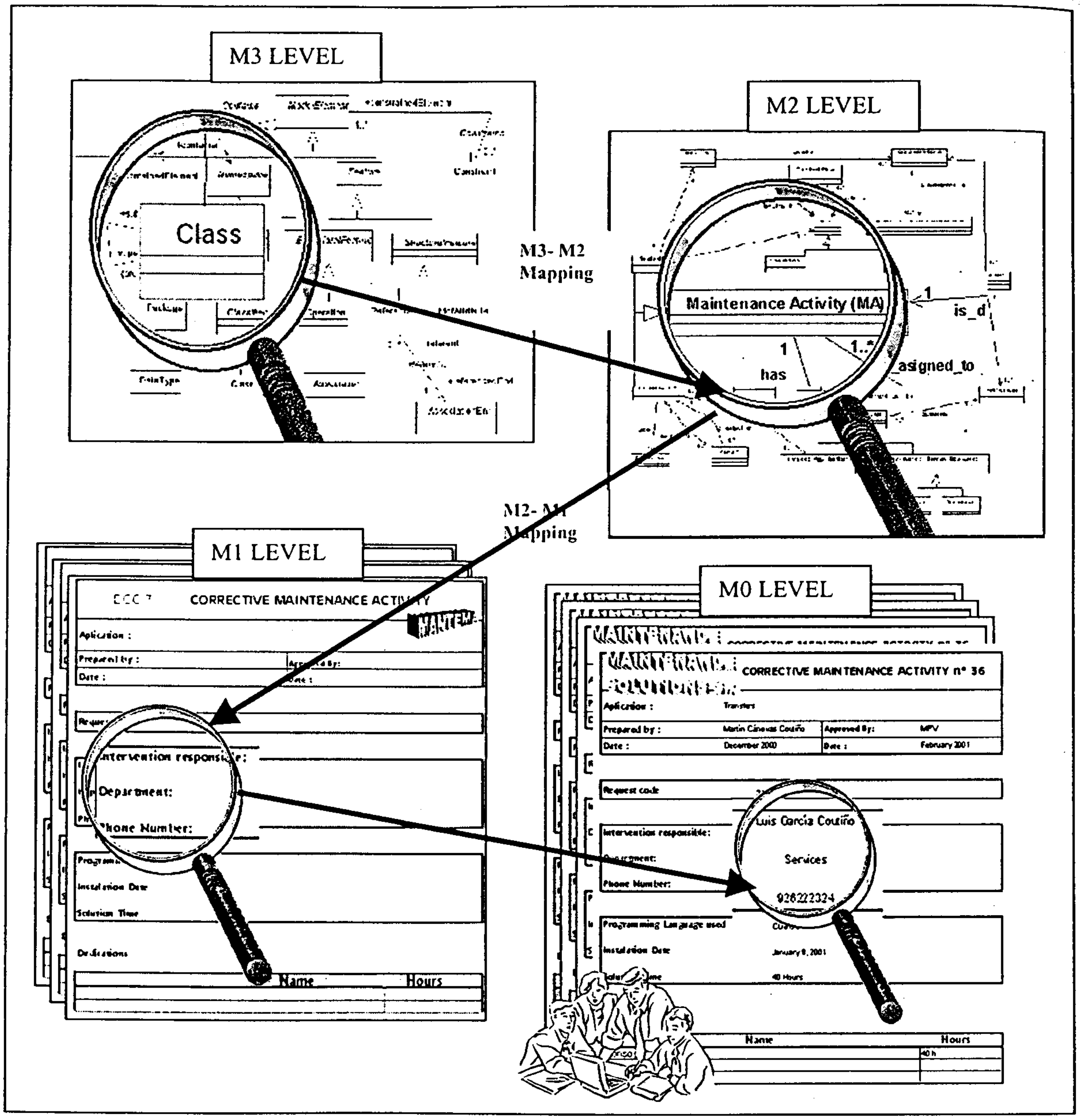

Figure 1- Example of MOF levels in MANTIS

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 modeling 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.

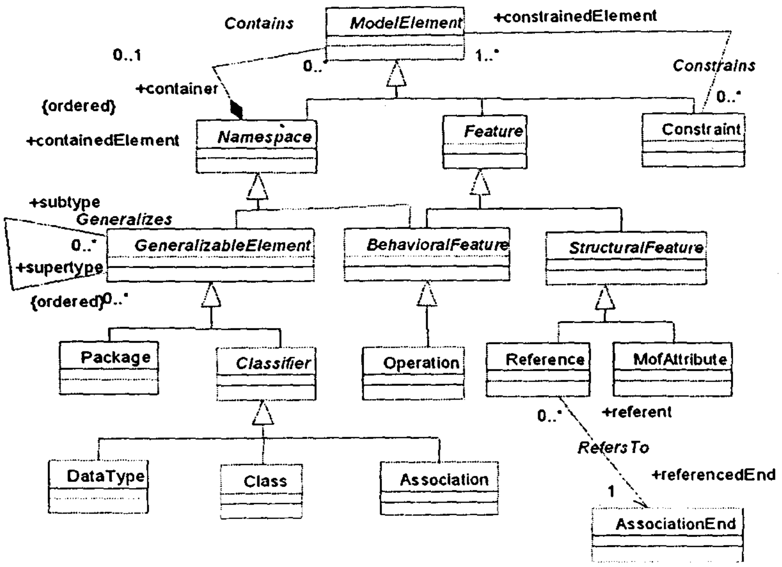

Figure 2- MOF Class Hierarchy.

By incorporating this last conceptual level and using standards for metamodeling (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 A Tool for software process metamodeling.

In this paper we present MANTIS-Metamod, a tool for representing and managing software process metamodels based on the MOF standard proposed by the Object Management Group [6]. This standard defines a framework using the principles of object oriented modeling for the definition of information models for metadata.

The tool provides support for the modeling 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 modeling of the SMP.

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

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.

For the realization of the aplication we have used a three-tier architecture [4] with the aim of reducing complexity and providing a degree of encapsulation. These vertical tiers are: presentation, application logic and storage.

In the following figure we can observe the three-tier Mantis-Metamod architecure:

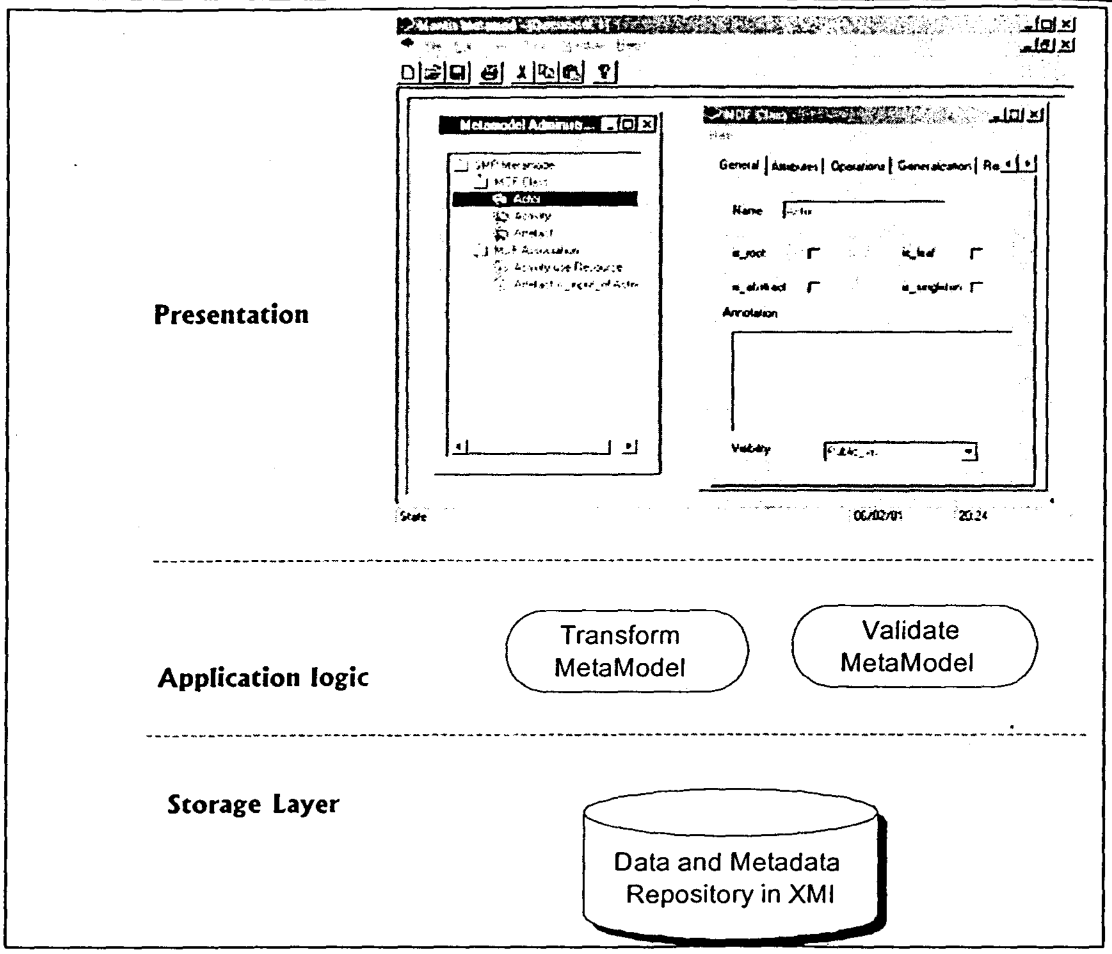

Figure 3- Three tier architecture of MANTIS-Metamod

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.

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. The use of XMI facilitates and permits the interchange of -etamodels based on MOF so the tool must 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 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.

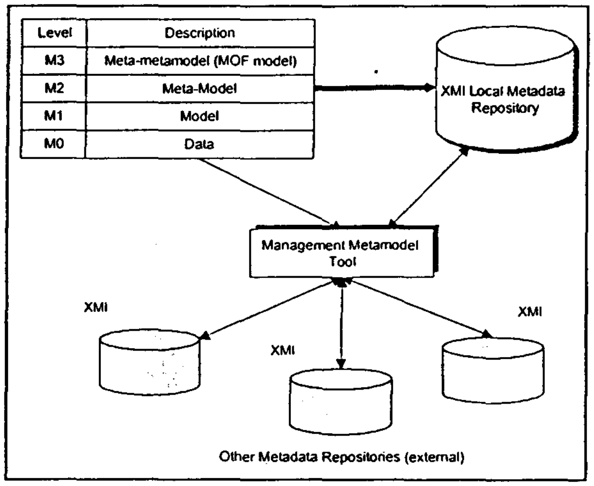

Figure 4- Application View.

As shown in figure 4 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 modeling 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 modeling the process and managing the different instances of execution (specific projects).

In this paper we have presented the importance of establish a conceptual level architecture for software process metamodeling. As example of this we have considered MANTIS, an integrated environment for the management of the maintenance software process.

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.

In order to provide automated support for the software process metamodeling, and in particular for the SMP, we have made 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.

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 modeling 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., y Webby, R., A Comprehensive Schema Integrating Software Process Modeling and Software Measurement, Fraunhofer Institute, IESE report Nº 047.99/E. v1.2, 1999.

[2] ISO/IEC 12207: Information Technology – Software Life Cycle Processes, 1995.

[3] ISO/IEC JTC1/SC7/WG4 15940 working draft 5: Information Technology – Software Engineering Environment Services, Juny-2000.

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

[5] OMG XML Metadata Interchange (XMI), v. 1.1, Oct-1999.

[6] OMG Meta Object Facility (MOF) Specification, v. 1.3 RTF, Mar-2000. In http://www.omg.org.

[7] Penadés, M.C.; Canós, J.; Carsí, J.A.; Hacia una Herramienta de Soporte al Proceso Software basada en la tecnología de Workflow. IV Jornadas de Ingeniería del Software y Bases de Datos; Cáceres, España, 1999.

[8] Piattini, M.; Ruiz, F.; Polo, M., Bastanchury, T., Fernández, I., Martínez, M.A. Mantenimiento del Software: Conceptos, Métodos, Herramientas y Outsourcing. Ed Ra-Ma, España 1998.

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

[10] 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.

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

[12] WfMC TC00-1003 1.1: Workflow Management Coalition. The Workflow Reference Model, Jan-1995.