
Applying a framework for the improvement of software process maturity



Gerardo Canfora¹, Félix García^{2,*},†, Mario Piattini², Francisco Ruiz² and C. A. Visaggio¹

¹*RCOST—Research Centre on Software Technology, Department of Engineering, University of the Sannio, Palazzo ex Poste, viale Traiano, 82100 Benevento, Italy*

²*Alarcos Research Group, University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain*

SUMMARY

This article presents the results and lessons learned in the application of the Framework for the Modelling and Measurement of Software Processes (FMESP) in a software company dedicated to the development and maintenance of software for information systems. The aim of FMESP is to provide companies with a conceptual and technological framework for the management of their process models and measurement models in an integrated way. Modelling and measurement are two key factors to promote continuous process improvement. As a result, important benefits were obtained. The company improved the maturity of its processes which allowed it to obtain the ISO 9000 certification. From a research point of view, Action-Research was successfully applied and as a result the framework was improved and important feedback was obtained, bringing to light new important issues which will be tackled in future work. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: software process modelling; measurement; improvement

INTRODUCTION

Software process improvement has received a great deal of attention in the last few years. As a matter of fact, continuous software process improvement is a fundamental goal for organizations which want to reach high levels of maturity. To support software process assessment and improvement, numerous initiatives have proposed reference models. Among these initiatives, the most widespread

*Correspondence to: Félix García, Alarcos Research Group, University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain.

†E-mail: felix.garcia@uclm.es

Contract/grant sponsor: MAS; contract/grant number: TIC 2003-02737-C02-02

Contract/grant sponsor: MECENAS

are Capability Maturity Model for Software (SW-CMM) [1], Capability Maturity Model Integration (CMMI) [2] and Software Process Improvement and Capability Determination (SPICE), which has been standardized as ISO 15504 [3].

Nowadays, it is common for customers to demand companies to be certified according to quality standards such as ISO 9000:2000 [4] or that they have high maturity levels in their processes in accordance with assessment models such as CMMI or ISO 15504. For this reason, software organizations are increasingly focusing on the management of their processes.

Management of software processes requires that four key responsibilities be assumed [5]: definition, measurement, control and improvement of processes. It is very important to consider the integrated management of two key aspects.

- *Process modelling.* Process models constitute the starting point for analysing, enacting and improving processes. Bearing in mind that there is a huge diversity among existing process modelling proposals, a reference process metamodel should be helpful. The Object Management Group has recently proposed the Software Process Engineering Metamodel Specification (SPEM) metamodel [6], which defines a reference language for process modelling.
- *Process evaluation.* The first step in software processes improvement is its evaluation. Due to the vast diversity of elements which could be measured in the context of software process evaluation (process models, projects and products), the establishment of an homogeneous terminology for the definition, calculation and exploitation of metrics is needed for the integrated and effective management of the measurement process.

In this paper we present the results and lessons learned from the application of the Framework for the Modelling and Evaluation of Software Processes (FMESP) in a software company. The FMESP framework includes the conceptual and technological elements which are needed to facilitate the management of the definition and evaluation of software processes.

The rest of the paper proceeds as follows. We first describe the characteristics of the framework FMESP. Then we present a general overview of the software company in which FMESP was applied. This is followed by a detailed description of the application of FMESP in the software company. Next we highlight the benefits for the company and the lessons learned. The final section outlines conclusions and further work.

FMESP

FMESP was developed in order to provide conceptual and technological support for the modelling and measurement of software processes in order to promote their improvement.

The research method used for the definition, refinement and application of FMESP was Action-Research (A-R). A-R is a collaborative research method which aims to merge theory and practice. A-R is focused on building new knowledge through the identification of solutions to real problems [7]. This is achieved by the intervention of researchers in the real world of practitioners and the results of this experience must be equally beneficial to both groups (see, for example, [8]).

The application of A-R was based on continuous feedback between the researchers and the company. As a result, a solution was found for the improvement of software process management in the company.

This solution consisted of managing, in an integrated way, the modelling and measurement of software processes, taking into account that it is not possible to carry out the measurement of software processes if a suitable process definition has not been previously established. From a measurement point of view, the following entities were identified as candidates for measurement: process models, projects and products. The measurement of projects and products might be easily supported given the vast diversity of proposals which exist in the literature, but we noted the non-existence of proposals related to the measurement of process models, and for this reason a set of metrics was defined and empirically validated [9].

Consequently the FMESP framework was developed and it is basically composed of two elements, namely a *Conceptual Framework* and a *Software Engineering Environment (SEE)*.

FMESP conceptual framework

The FMESP Conceptual Framework, largely described in [10] provides the necessary support for the representation and management of the knowledge related to software processes from the modelling and measurement perspectives, by integrating the two. Figure 1 illustrates the structure and main elements of the conceptual framework.

- (a) A *conceptual metadata architecture of four abstraction layers*. This architecture realizes the integration of modelling and measurement by means of the representation of the elements in different abstraction layers.
- (b) A collection of *ontologies*.
 - *Descriptive software process modelling ontology*. This ontology was defined to clarify the domain of the descriptive modelling of software processes and the SPEM specification [6] was used as a reference.
 - *Software measurement ontology* [11]. With this ontology the software measurement domain was clarified by identifying its existing concepts and relationships. As a result, a consistent measurement terminology was provided which enabled us to consequently define a suitable measurement metamodel.
- (c) A collection of *metamodels*. The conceptual architecture includes the following metamodels.
 - A *process modelling language (PML)*. The PML considered in the framework was SPEM.
 - A *software measurement metamodel*. We defined a measurement metamodel based on the software measurement ontology.
 - *Metamodels* for the definition of the *software-process related entities*. These metamodels, called 'domain metamodels', represent the kind of entities which are candidates for measurement in the context of software process evaluation.

FMESP SEE

The *FMESP SEE* comprises two integrated tools which provide the technological support for the conceptual framework: *METAMOD* for the definition and management of the metadata conceptual architecture; and *GenMETRIC* to support the development of measurement models. In Figure 2, a general overview of the SEE is shown.

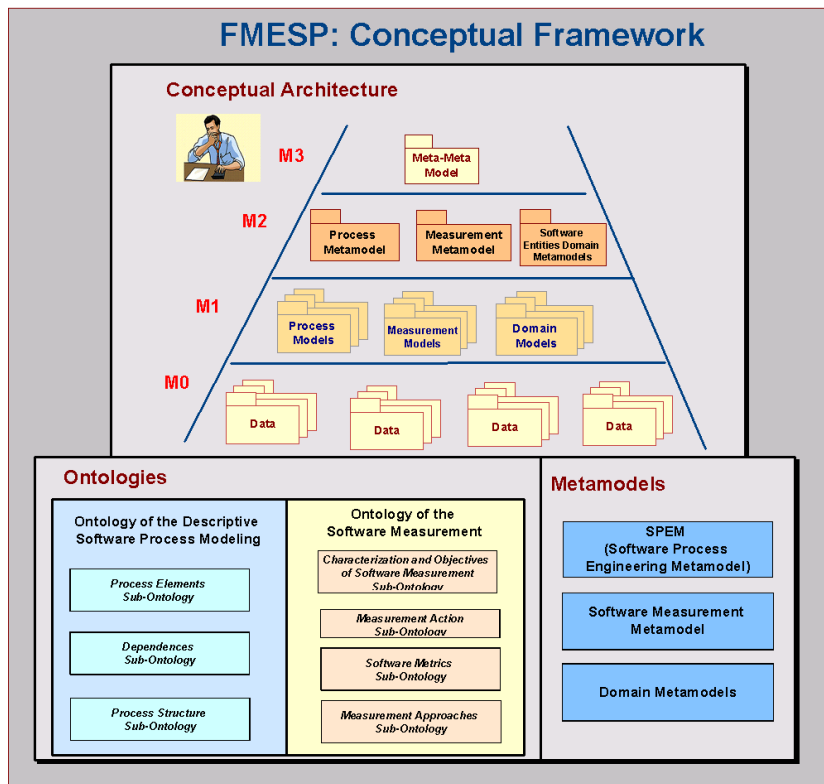


Figure 1. FMESP Conceptual Elements.

- (a) *A metadata repository.* The repository is the key component of the SEE and it provides the tools for metadata management, including the derivation of concrete models. The models and metamodels are stored in the repository as XMI documents [12].
- (b) *Software components for the management and storage of metadata.*
- *MOImplementation* manages the representation of metamodels and models in accordance with the MOF [13] standard.
 - *RepManager* [14] deals with the XMI representation and the storage of metadata.
- (c) *Software tools.*
- *METAMOD* [15] supports the definition, query and storage of the models and metamodels of the repository by using the services of the components *MOImplementation* and *RepManager*.

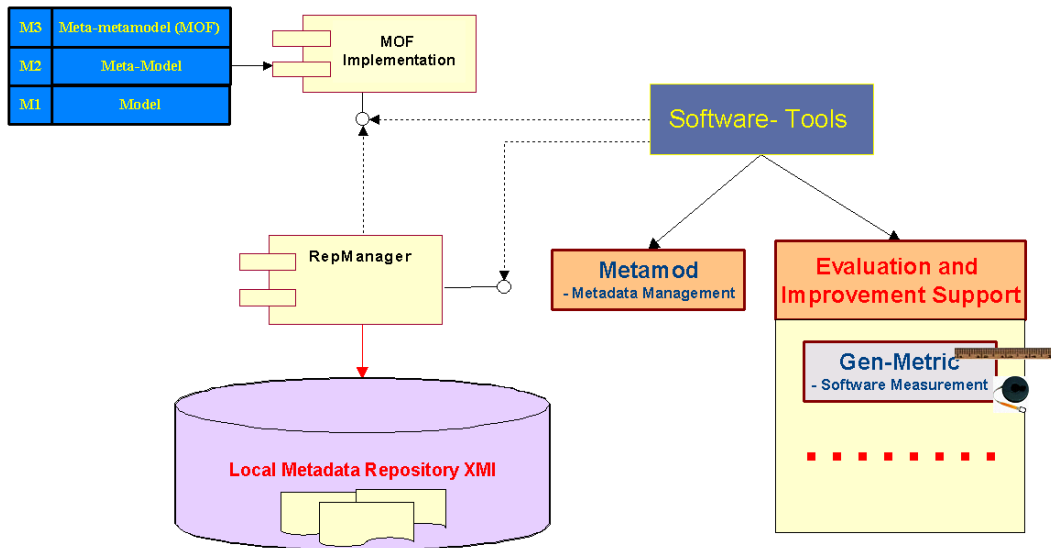


Figure 2. FMESP SEE.

- *GenMETRIC* [16] supports the measurement of software entities. GenMETRIC permits the measurement of any software entity, which must be stored in the repository, in accordance with the domain metamodel's definition. Furthermore, the definition of any software metric is also possible.

With the proposed framework, the management of the modelling and measurement of software processes is supported by including a metadata repository in which a knowledge base concerning the process is established for its evaluation and improvement.

DESCRIPTION OF THE COMPANY

FMESP was applied to a Spanish company, Cronos Iberica S.A., whose main activity is the development and maintenance of software for information systems. A summary of its characteristics is provided in Table I.

FMESP was applied in order to provide the necessary support to satisfy the quality goals of Cronos Iberica, especially with regards to process improvement.

FMESP: CASE STUDY

In this section we describe how FMESP was applied to Cronos Iberica in order to improve its software. The steps followed in the application are represented in Figure 3.

Table I. General characteristics of the company 'Cronos Iberica'.

Creation date	May 1989
Number of employees	~100
Main areas of professional activity	<ul style="list-style-type: none"> • Customized development of software applications for public administration and the private sector • System plans and strategic consultancy • Database development
Quality policy	<ul style="list-style-type: none"> • To apply the necessary means to assure that products and services satisfy the requirements and expectations of customers • To develop information systems following all their lifecycle phases, in accordance with established schedules and costs • To promote continuous improvement

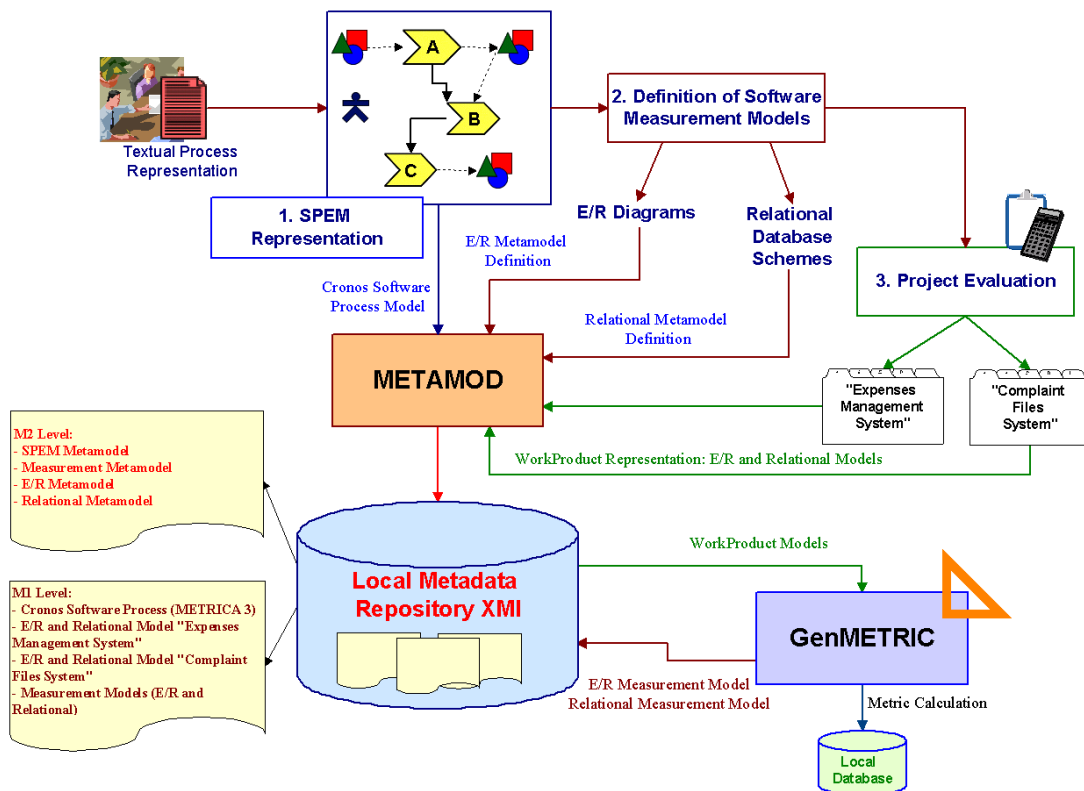


Figure 3. FMESP application in Cronos Iberica.

1. *Process representation.* First of all, the SPEM metamodel was used for the representation of the software processes of Cronos Iberica in order to facilitate its understanding, evaluation and automated support.
2. *Definition of measurement models.* After the definition of the software processes of Cronos, the relevant entities related to these processes and candidates for measurement (E/R diagrams and relational schemes) were identified. Measurement models of these entities were defined to satisfy the information needs of the company.
3. *Project evaluation.* Finally, two projects carried out in Cronos (in accordance with its software processes), were evaluated. This evaluation consisted of the application and calculation of the metrics included in the measurement models; these models were defined relying on the relevant work products (E-R model and relational database scheme) produced by these projects.

A more detailed description of the steps carried out in the application of FMESP is provided in the following subsections.

Process representation with SPEM

The first step in the FMESP application was the representation using the SPEM metamodel of the software processes of the company. Cronos Iberica has a defined software process for the development of its professional activities. The software process of Cronos is based on the Spanish public methodology METRICA v3.0 [17], but it is particularly adapted to the development of database applications. The main process is 'Development', in which the technical processes related to product development, from requirement management to the installation and delivery to the customer, are included. Apart from the main development process, two supporting processes were defined: 'project planning' and 'configuration management'.

The SPEM definition of the processes was performed by examining the existing textual documentation and after some meetings with the project managers of the company in which some doubts related to the documentation (deriving from existing inconsistencies) were resolved and new knowledge which was not included in the documentation (but was considered in the execution of the projects) was obtained. To provide a graphical representation of the processes, the UML class, state chart and activity diagrams stereotyped with SPEM were used. In Figure 4, the activity diagram developed to represent the activity 'Analysis and Design' (belonging to the 'Development' process) is shown and in Figure 5 the diagram class which defines the structure of the work product 'analysis document' is represented.

Measurement model definition

After the SPEM representation of Cronos's software processes, the software measurement models were defined. Two work products were chosen as the most relevant to be measured: conceptual and logic data models, which were represented by using entity/relationship diagrams and relational database schemes. These artefacts were chosen because the main professional activity of Cronos is the development of database applications.

The metric definition was performed by analysing the metamodels which represent the process entities (domain metamodels). This analysis is based on the definition of the direct metrics based on the

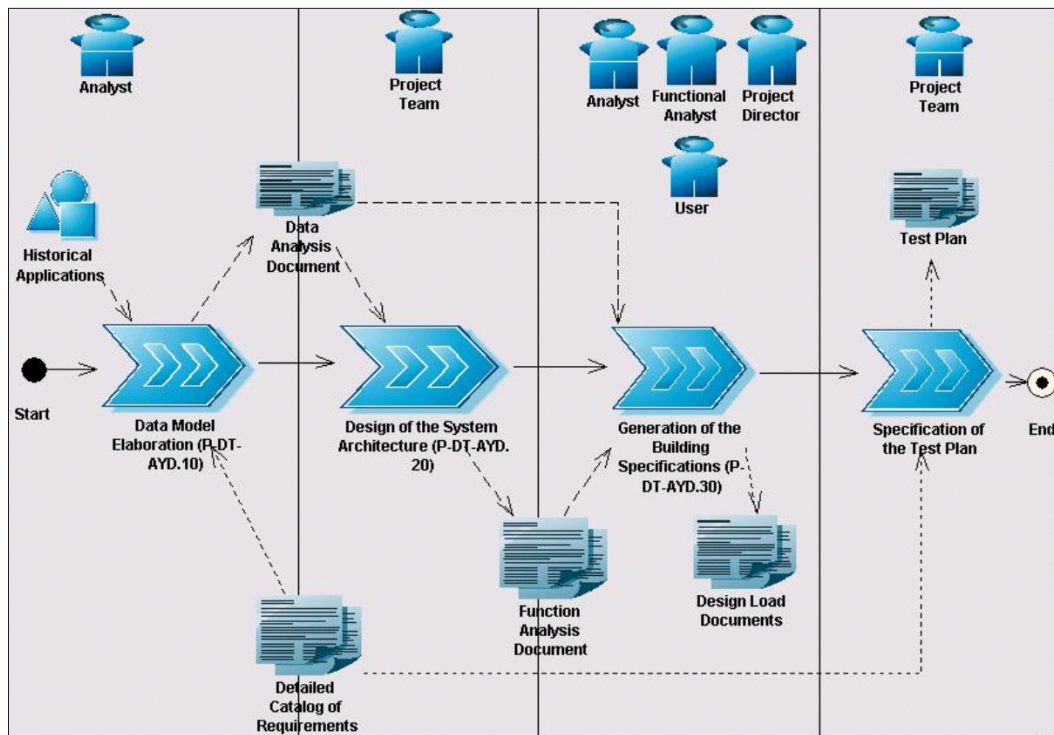


Figure 4. SPEM representation of the activity 'analysis and design'.

classes (constructors) and relationships included in the metamodel. There were no explicit metamodels in Cronos to represent their conceptual and logical data models, e.g. the E/R metamodel and the relational metamodel. These metamodels were defined and incorporated into the XMI repository by using the METAMOD tool. In Figure 6 the UML graphical representation (MOF compliant) which represents the E/R metamodel elements and their relationships is shown.

The next step was the definition of the measurement models: we established which 'information needs' had to be satisfied by the measurement of the relevant work products. One significant requirement of the company was to know how easy it would be to maintain these artifacts, given the need to maintain the database applications, especially those developed for Spanish Public Administrations (who are very important customers of the company). These kinds of applications are very likely to evolve due to changes in regulations, new laws, etc. To satisfy information needs, two proposals of metrics to evaluate the maintainability of conceptual and logical data models were considered [18,19] and the metrics were included in the measurement models.

The model definition was performed according to the measurement metamodel and GenMETRIC was used to perform the model definition. In Appendix A, the measurement model defined for the

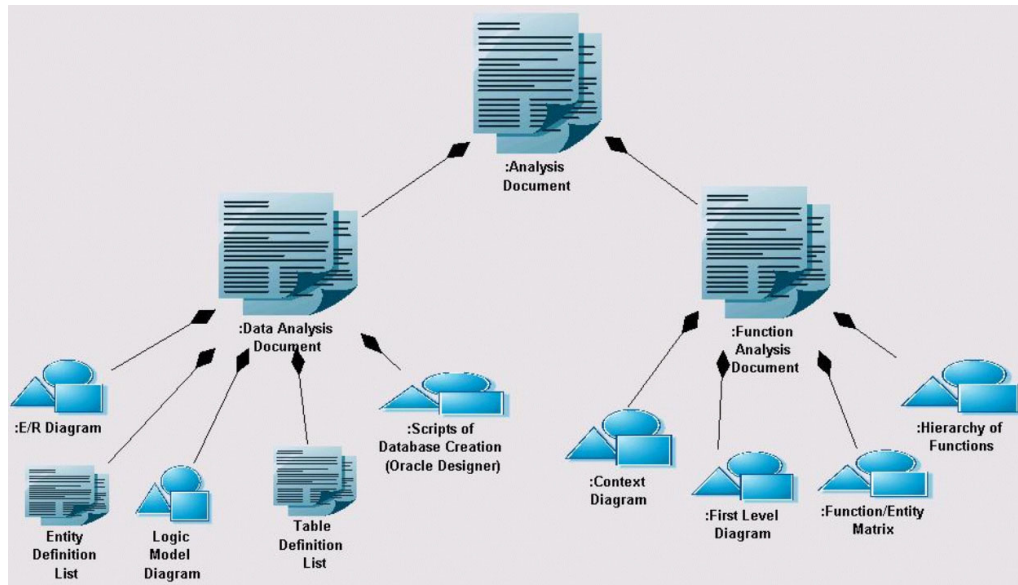


Figure 5. SPEM representation of the work product structure 'analysis document'.

E/R diagrams is shown. The metrics used for the measurement models of E/R diagrams are those which were proposed by Genero *et al.* [18]. According to the FMESP measurement metamodel, metric definition is performed in the following order depending on the kind of metrics (see Appendix A):

- *Direct metrics* were defined on the domain metamodel which represents the software entity. For example, the direct metric 'number of entities' is defined on the E/R metamodel constructor 'entity' by applying the *measurement method* 'count'.
- *Indirect metrics* are calculated by applying a *measurement function* on other direct/indirect metrics. For example, the number of relationships in an E/R diagram (NNR) is calculated on the metrics NBinaryR (number of binary relationships) and NN-AryR (number of n-ary relationships) by applying the function: $NNR = NbinaryR + NN-AryR$.
- *Indicators* were obtained by applying an *analysis model* on other direct and/or indirect metrics. For example, the indicator Entity Maintenance Index (IME) was defined as the proportion of entities and attributes of the diagram and some decision criteria were defined. Before the application of the FMESP framework, Cronos had defined a set of project indicators. The experience of the company in the development and maintenance of database applications and in the definition of indicators was required to establish the decision criteria associated with the indicators defined in the measurement models for E/R and relational artefacts. For example, for the IME indicator the following decision criteria were used by the company experts: If $IME > 15$ then $IME = \text{'very high'}$; 'If $10 < IME \leq 15$ then $EMI = \text{'high'}$ and so on.

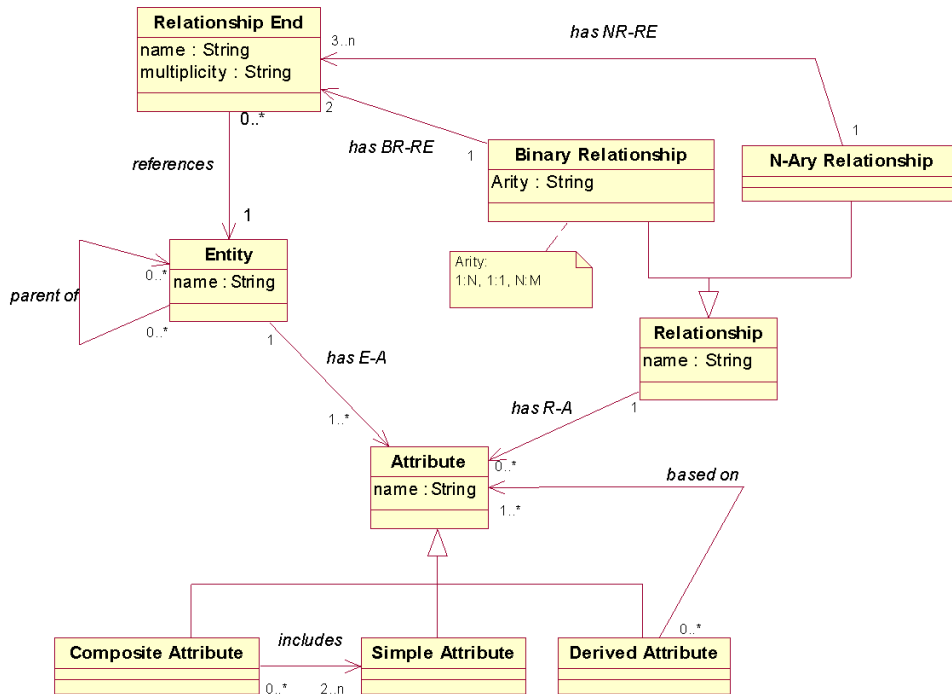


Figure 6. E/R metamodel defined.

Using the measurement metamodel of FMESP, Cronos obtained a consistent representative metamodel together with the following: the existing, but not entirely documented, indicators, the satisfied information needs of the company, the metrics (direct and indirect) they referred to, the associated decision criteria, etc. Additionally, Cronos performed the measurement of these indicators which were based only on project information, but product measurements were not carried out. The same steps were followed in order to define the measurement model for the relational schemes.

After the definition of the measurement models, the evaluation of the project took place. This evaluation consisted of calculating the metrics (direct, indirect, indicators) defined in the measurement model on the E/R diagrams and relational schemes obtained in the projects considered in the case study.

Project evaluation

After having defined the domain metamodels of the E/R diagrams and relational schemes and the measurement models of these entity types, two small projects were evaluated (see Table II).

With regard to the application of FMESP, the (conceptual and logical) data models obtained as a result of these projects were evaluated. The measurement was based on the calculation of the metrics

Table II. Projects evaluated.

	Expenses Management System	Complaint Files Management System
Client	Spanish Public Administration*	
Start date	14 October 2002	31 March 2003
End date	30 December 2002	3 July 2003
General description	Expenses management (authorizations, costs and expenses) of the administration personnel	Management and tracking of European Commission files relative to infractions
Development environment	Oracle/Developer 8	Oracle/Developer 8
Development type	Adaptation and functional enhancement of a former system developed with 'Clipper' and 'Dbase'	
Subsystems	Trips Table maintenance Administration Report	Complaints Table maintenance Report Statistics

*Name not provided to preserve the confidentiality.

with the aim of satisfying the information needs in accordance with the measurement models defined. The tool GenMETRIC was used for the automatic calculation of the metrics.

With regard to the evaluation of the E/R diagrams, the E/R diagrams obtained as instances of the defined E/R metamodel were represented. Figure 7 illustrates the E/R diagram obtained in the 'Expenses Management System' (Oracle Designer Notation).

The definition of the E/R model as an instance of the 'E/R metamodel' was performed considering the graphical representation of the model (Figure 7) and the entities list, which is another work product of the Cronos software process. The Metamod tool was used to carry out such a definition (Figure 8). At the time of the FMESP application, a tool did not exist for the definition of E/R diagrams with XMI exportation capability, and for this reason we had to define the E/R metamodel.

Table III gives the results for the measurements carried out in accordance with the defined E/R measurement model (see Appendix A). This measurement model belongs to the E/R diagram of the 'Expenses Management System' project and was supported by GenMETRIC.

The measurement metamodel ('Measurement Action' package) was used to represent, in a consistent way, the results of the measurement. The measurement of the relational schemes of the projects evaluated was performed in the same way.

The same process was followed in order to measure the 'relational databases' by defining the relational metamodel, the measurement model (which was based on the metrics proposed by Calero *et al.* [19]) and the evaluation of the relational schemes obtained as a result of the projects considered.

By using the FMESP framework for performing the measurement process, the results are not registered in an isolated way, but are related with all the necessary metadata (attributes, metrics, information needs, tools, etc.). This constitutes a suitable and necessary context for performing the measurement process in an effective way.

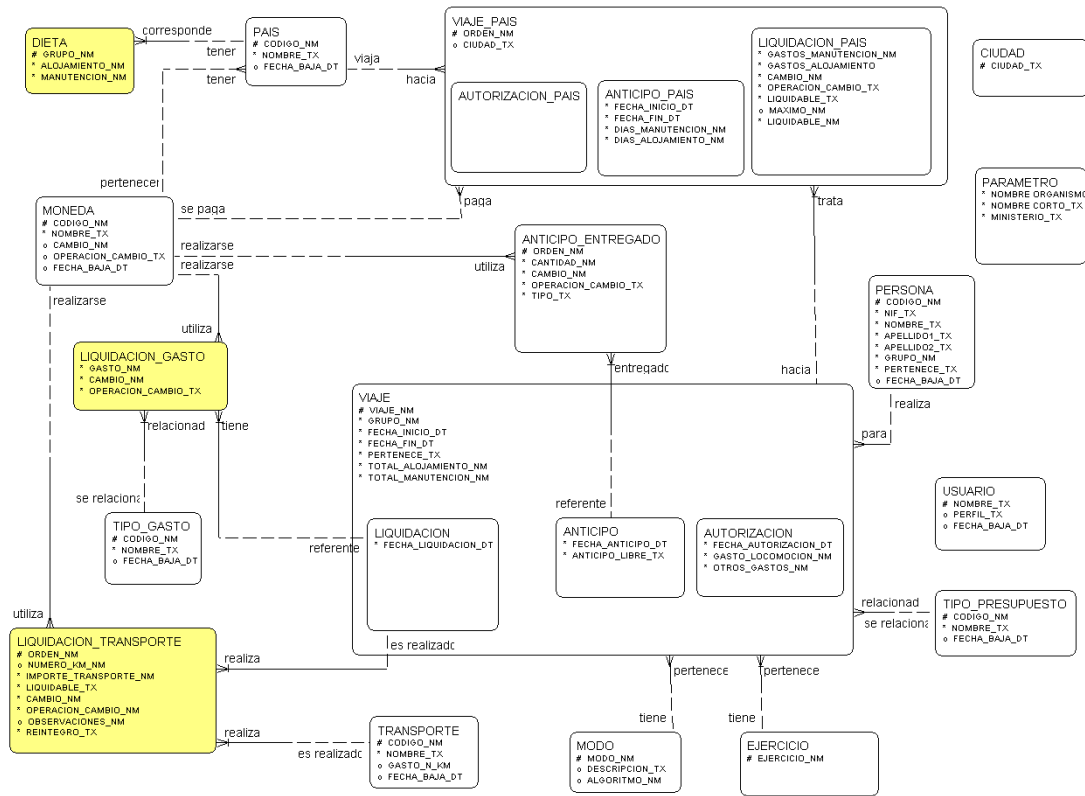


Figure 7. E/R diagram of the project ‘Expenses Management System’ represented using the Oracle Designer.

CONTRIBUTIONS AND LESSONS LEARNED

The lessons learned as a result of the utilization of FMESP can be tackled from two different points of view: company and researchers.

From the *company perspective*, the main contributions provided can be listed as follows.

Software process modelling

- The SPEM process representation provided Cronos with an efficient and homogeneous definition of its own processes, relying on a consistent and simple terminology. In addition, the existing textual documentation was complemented with a graphical view which facilitates the correct understanding of the processes. Adaptation and modification of the defined processes and improvement opportunities are thus eased.

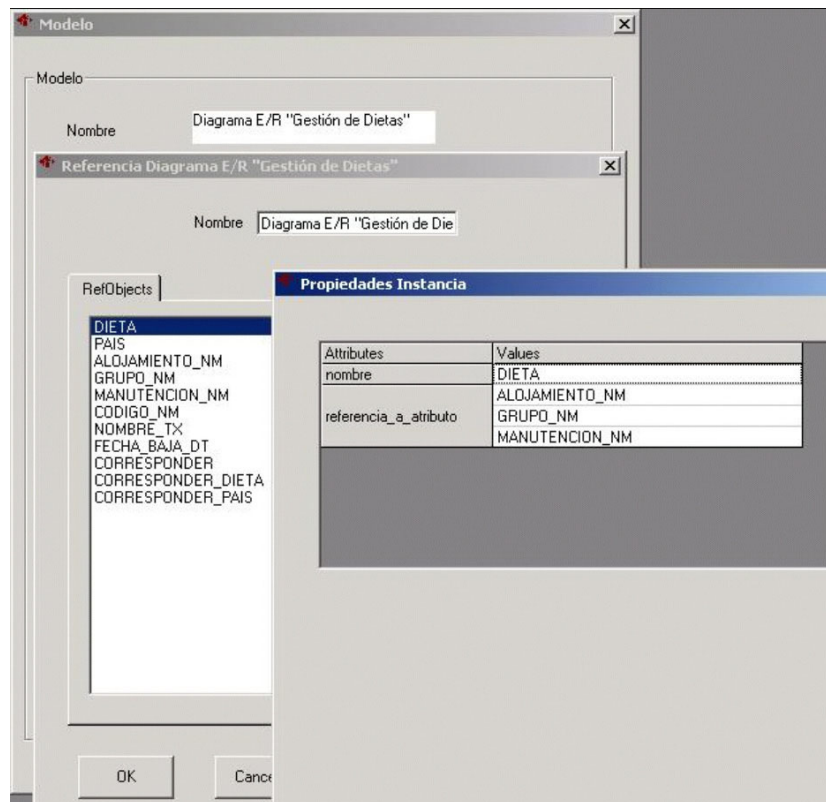


Figure 8. Definition with METAMOD of the E/R diagram obtained in the project 'Expenses Management System'.

- The use of SPEM as a PML required a systematic definition of the processes which was not provided by a pure textual representation. As a result, some faults and inconsistencies were detected and removed and also additional process information (some activities, work products, roles and dependences among them) was explicitly represented. This additional information was not registered but the people responsible took it into account when carrying out the processes. This can facilitate the awareness and consciousness of the process.
- Cronos's software processes were represented in an automated way which allowed the company to develop tools for the management of these processes. Previously, any process change was included in an 'ad hoc' manner by modifying textual documents. With the assistance of proper software tools these tasks could be enormously facilitated and improved.

Table III. Measurement results of the E/R diagram obtained in the 'Expenses Management System' project.

Measurement	Attribute	Metric	Measure
1	Size	NE	23
2	Size	NAS	82
3	Size	NCA	0
4	Size	NDA	0
5	Complexity	NM:NR	0
6	Complexity	N1:NR	17
7	Complexity	NbinaryR	17
8	Complexity	NN-AryR	0
9	Complexity	NIS_AR	6
10	Complexity	NrefR	0
11	Size	NA	82
12	Complexity	NNR	17
13	Size	IME	3,56 ('Low')
14	Complexity	IC	0,74 ('Low')
15	Complexity	IH	0,26 ('Medium')
16	Complexity	INM	0 ('Low')
17	Complexity	CI	0 ('Low')

- The current representation facilitates the evolution of processes and it will be very advantageous for the company, especially at this particular moment when it is adapting its software processes in order to support (in its database development methodology) three-tier client–server architectures.

Measurement

- A consistent and unique terminology is provided in order to manage and carry out the measurement process in an effective way.
- The former measurement system of Cronos was based only on the calculation of some project indicators. Now, it has been complemented and improved with a more complete measurement template. The current measurement process in Cronos has been enriched with information and tools for computing and interpreting the indicators.
- The measurement metamodel applied is based on ISO/IEC 15939 [20], which provides Cronos with a measurement process compatible with the most representative measurement standard. Standards provide organizations with agreed and well-recognized practices and technologies, which assist them to interoperate and to work using engineering methods, reinforcing the software engineering as an 'engineering' discipline, instead of a 'craft' [21]. With a successful measurement process, the measurement is embedded in the organization and performance improves because people are making fact-based decisions [22].
- Two measurement models (E/R diagrams and relational databases) have been provided and as a result Cronos currently has two consistent models to measure the most relevant work products

obtained in their processes in an integrated and systematic way. This is very important as it allows Cronos to guarantee the quality of its software products according to its policy.

Technological support

- A SEE composed of two software tools has been provided to support the process definition and measurement.
- The SEE can be easily extended with new tools to automatically manage aspects related to process improvement.
- This SEE constitutes a starting point for the development of other tools in order to completely install the technological framework of FMESP in an industrial environment.

As a result of the quality policy of the company and the contributions of FMESP, the company has improved the maturity of its processes. This improvement has been shown in the following aspects.

- The company has obtained the ISO 9001:2000 quality certification. Taking into account the existing correlation between maturity models as CMM or CMMI and the ISO 9000:2000 standard [23,24], Cronos could be considered to reach a maturity level between levels 2 and 3 of CMMI.
- In addition, the application of the FMESP framework provides the company with the necessary support to fulfil the requirements of higher maturity levels, namely:
 - process definition support: the FMESP framework provides the means necessary to satisfy the general goal of a CMMI level 3: '*Institutionalize a defined process*'.
 - process measurement support: the FMESP framework supports the measurement of projects, products and processes by providing a suitable measurement metamodel (CMMI, ISO 15939 and PSM compliant [22]) and a flexible method to measure these kinds of artifacts at metamodel scope; it provides companies with the initial support necessary to sustain the measurement process, thus enabling satisfaction of the CMMI measurement-related key process areas (levels 3, 4 and 5) [25].

From the *research perspective*, this experience has allowed us to validate and refine the proposal by successfully applying the A-R method. It has allowed us to confirm the importance of the following aspects.

- *Rigorous definition of the software processes* as a prerequisite for their evaluation and improvement.
- *Effective and integrated representation of the measurement process related information* by means of a suitable metamodel based on a consistent terminology, given the diversity of process-related entities that are candidates for measurement.
- *Application of the measurement process* not in an isolated way, but *integrated in the context of evaluation and improvement of software processes* by identifying the relevant entities which are necessary to measure and registering the results obtained in a homogeneous and consistent way.
- *Suitable software tools* to automate and manage the software processes in order to promote their improvement. The FMESP application has allowed the researchers to refine the developed software prototypes.

CONCLUSIONS AND FUTURE WORK

In this paper we have presented an experience in which the framework FMESP was applied in a software company in order to improve their software processes. This framework integrates the modelling and measurement of software processes by providing the conceptual and technological support necessary to successfully manage these two key responsibilities of the process management. As a result of the application of the framework, important benefits were obtained.

From an industrial point of view, the support for the consistent representation of software processes was provided and Cronos's processes were represented with SPEM. One of the most relevant benefits of FMESP is to provide a consistent and effective way to perform the measurement process. The metrics (direct and indirect) applied in the case study were previously validated empirically through families of experiments in which students and professionals were involved, as described in more detail in [19,26], and following the method proposed in [27]. Therefore, as a result, the company applied in the real environment a set of metrics the utility of which was previously validated. Additionally, application of the measurement models for E/R and relational schemes in Cronos provided useful feedback in order to refine and improve the metrics and indicators defined.

The framework has also enabled Cronos to obtain the ISO 9000 quality certification. In addition, it provides Cronos with the needed support to reach higher maturity levels according to models such as CMMI.

From the research perspective this experience has contributed to validating the FMESP proposal, by providing important evidence of its practical usefulness. Furthermore, some important ideas and improvements for the future have been discovered.

- A fundamental issue to consider will be the support necessary to refine the base and derived measures defined for maintainability of the E/R and relational schemes. This refinement will be based on the results of applying the measures in maintenance projects of database applications in Cronos. Besides, refinement of the values of the decision criteria associated to the defined indicators and the proposal of new indicators will be tackled in order to facilitate decision making.
- Development of new measurement models for object-oriented work products (based on the metrics proposed by Manso *et al.* [28]) and for data warehouse models (metrics proposed by Serrano *et al.* [29]) to complete the integration of the measurement of the work products, especially given the current adaptation of Cronos's processes to object-oriented technologies.
- Carrying out the required actions to continue to improve the process maturity level of Cronos.

Other relevant future works include the following.

- Measurement metamodel: it would be useful to develop a graphical notation for the representation of measurement models according to the measurement metamodel of FMESP. As a result, a software tool which extends the functionality of GenMETRIC should be incorporated into the SEE.
- SEE: the tools developed must be refined in order to develop a more robust environment which can be used in an industrial setting.
- Apply the FMESP framework in more cases in order to confirm its advantages.
- Software process improvement based on knowledge management: one important future area of research is knowledge management application to promote software process improvement [28].

The FMESP framework is suitable for this application because it provides an open repository in which the metadata relating to the definition and measurement of software processes are included. This information can be used by a system of intelligent agents in order to derive improvement actions. We are currently defining a prototype for this [30].

In summary, if an organization wants to effectively manage its processes, it needs to define its processes in an understandable and homogeneous way and to collect and analyse the different metrics necessary in an integrated way. Otherwise the company would have tools and techniques to evaluate the aspects of the process separately. FMESP provides the necessary tools for integration and support. The suitability of this framework is reinforced by the current trend for ‘model-based process engineering’ [31] in order to manage the special complexity of software processes.

APPENDIX A. MEASUREMENT MODEL FOR E/R DIAGRAMS

Table AI. E/R measurement model. Instance of the package ‘Characterization and Objectives’.

Measurement metamodel elements (MOF classes)	E/R measurement model (instances M2)		Measurement metamodel relations (MOF associations)	E/R measurement model (links M2)	
Attribute	Size	Complexity	Includes (subentity)	Entity class	—
Entity class	E/R diagram		Defined for	Quality model Entity class	ISO 9126 E/R diagram
Measurable concept	Maintainability		Evaluates	Quality model	ISO 9126
Entity	E/R diagram of the ‘Expenses Management System’		Belongs to	Measurable concept Entity	Maintainability
	E/R diagram of the ‘Complaint Files Management System’			Entity class	E/R diagram of the ‘Expenses Management System’ E/R diagram of the ‘Complaint Files Management System’
Quality model	ISO 9126		Relates	Attribute Measurable concept	Size Complexity Maintainability
Information need	To know the E/R diagram’s maintainability		Is associated with	Measurable concept Information need	Maintainability To know the E/R diagram’s maintainability
			Submeasurable concept Has	Measurable concept Entity class Attribute	— E/R diagram Complexity

Table AII. Direct metrics and measurement methods defined to measure E/R diagrams.

Metric (Genero)	Description	Measurement method	Definition of the metric on the E/R metamodel (see Figure 6)
NE	Number of entities	'Count the number of entities of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Entity</i> (e.g. count the number of instances of this class in the E/R model)
NAS	Number of simple attributes	'Count the number of simple attributes of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Simple Attribute</i>
NCA	Number of composite attributes	'Count the number of composite attributes of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Composite Attribute</i>
NDA	Number of derived attributes	'Count the number of derived attributes of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Derived Attribute</i>
NM:NR	Number of M:N relationships	'Count the number of M:N relationships of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Binary Relationship</i> with the following constraint: the ' <i>Arity</i> ' property of the <i>Binary Relationship</i> has value 'M:N'
N1:NR	Number of 1:N relationships (including 1:1 relationships)	'Count the number of 1:N relationships of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Binary Relationship</i> with the following constraint: the ' <i>Arity</i> ' property of the <i>Binary Relationship</i> has value '1:N' or '1:1'
NBinaryR	Number of binary relationships	'Count the number of binary relationships of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Binary Relationship</i>
NN-AryR	Number of N-Ary (no binary) relationships	'Count the number of N-Ary relationships of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>N-Ary Relationship</i>
NIS_AR	Number of IS_A relationships (generalization/especialization)	'Count the number of IS_A relationships of the E/R diagram'	'Count' measurement method on the metamodel MOF-Association <i>Parent Of</i> (e.g. count the number of links of this association in the E/R model)
NrefR	Number of reflexive relationships	'Count the number of reflexive relationships of the E/R diagram'	'Count' measurement method on the metamodel MOF class <i>Binary Relationship</i> with the following constraint: 'the <i>Binary Relationship</i> has two <i>Relationship Ends</i> which reference the same entity'

Table AIII. Indirect metrics and measurement functions defined to measure E/R diagrams.

Metric	Description	Measurement function
NA	Total number of attributes in an E/R diagram, considering the entities and relationships attributes. This total includes simple, composite and derived attributes, each of them takes the value 1	$NA = NAS + NCA + NDA$
NNR	Total number of relationships in an E/R diagram (without considering IS_A relationships)	$NNR = NbinaryR + NN-AryR$

Table AIV. Indicators, analysis models and decision criteria to evaluate E/R diagrams.

Metric	Description	Analysis model	Decision criteria
IME	Entity maintenance indicator: ratio of attributes and entities of the diagram. The higher the value, the higher the maintenance difficulty	$IME = NA/NE$	<p>If $IME > 15 \rightarrow IME = \text{'Very High'}$ If $10 < IME \leq 15 \rightarrow IME = \text{'High'}$ If $5 < IME \leq 10 \rightarrow IME = \text{'Medium'}$ If $0 \leq IME \leq 5 \rightarrow IME = \text{'Low'}$</p>
IC	Diagram connectivity indicator: ratio of relationship and entities of the diagram. The higher the value, the higher the maintenance difficulty	$IC = NNR/NE$	<p>If $IC \geq 2 \rightarrow IC = \text{'Very High'}$ If $1,5 \leq IC < 2 \rightarrow IC = \text{'High'}$ If $1 < IC < 1,5 \rightarrow IC = \text{'Medium'}$ If $0,5 \leq IC \leq 1' \rightarrow IC = \text{'Low'}$ If $0 < IC < 0,5 \rightarrow IC = \text{'Very Low'}$</p>
IH	Diagram inheritance indicator: ratio of inheritance relationships and total number of relationships (including inheritance). The higher the value, the higher the maintenance difficulty	$IH = \frac{NIS_AR}{NNR + NIS_AR}$	<p>If $IH \geq 0,5 \rightarrow IH = \text{'High'}$ If $0,1 < IH < 0,5 \rightarrow IH = \text{'Medium'}$ If $0 \leq IH \leq 0,1' \rightarrow IH = \text{'Low'}$</p>
INM	N:M relationship indicator: ratio of N:M relationships. The higher the value, the higher the maintenance difficulty	$INM = NM : NR/NNR$	<p>If $INM \geq 0,5 \rightarrow INM = \text{'High'}$ If $0,25 < INM < 0,5 \rightarrow INM = \text{'Medium'}$ If $0 \leq INM \leq 0,25 \rightarrow INM = \text{'Low'}$</p>
CI	Relationship complexity: this indicator represents the ratio of N-Ary relationships. The higher the value, the higher the maintenance difficulty	$CI = NN-AryR/NNR$	<p>If $CI \geq 0,2 \rightarrow CI = \text{'Very High'}$ If $0,1 < CI < 0,2 \rightarrow IH = \text{'High'}$ If $0,05 < CI < 0,1 \rightarrow IH = \text{'Medium'}$ If $0 \leq CI \leq 0,05' \rightarrow IH = \text{'Low'}$</p>

Table AV. E/R measurement model. Instance of the package 'Metrics'.

Measurement metamodel elements (MOF classes)	E/R measurement model (instances M2)	Measurement metamodel relations (MOF associations)		E/R measurement model (links M2)		
Scale (type)	Integers from 0 to infinity (ratio)	Defined for	Metric	NE, NA, NAS NCA, NDA, IME	N1:NR NM:NR NBinaryR NNA-ryR NIS_AR NRefR NNR IC, IH, INM, CI	
			Attribute	Size	Complexity	
Metric (unit)	(see <i>direct metric, indirect metric and indicator</i>)	Transformation	Metric		—	
			Metric		—	
Direct metric (unit)	NE (entity) NAS (attribute) NCA (attribute) NDA (attribute) NM:NR (relationship) N1:NR (relationship) NbinaryR (relationship) NN-AryR (relationship) NIS_AR (relationship) NrefR (relationship)	Has	Metric	NE, NAS, NCA, NDA, NM:NR, N1:NR, NbinaryR, NN-AryR, NIS_AR, NrefR, NA, NNR	IH, INM, CI	IME, IC
			Scale	Integers from 0 to infinity	Reals between 0 and 1	Real from 0 to infinity
Indirect metric (unit)	NA (attribute) NNR (relationship)					
Indicator	IME (%) IC (%) IH INM CI					

Table AVI. E/R measurement model. Instance of the package ‘Measurement Approaches’.

Measurement metamodel elements (MOF classes)	E/R measurement model (instances M2)	Measurement metamodel relations (MOF associations)	E/R measurement model (links M2)		
Measurement method	See Table AII	Uses	Direct metric	See Table AII	
Measurement function	See Table AIII	Calculated with	Measurement method Indirect metric	See Table AIII	
Analysis model	See Table AIV	Uses	Measurement function Direct metric	$NA = NAS + NCA + NDA$ NAS, NCA, NDA	$NNR = NbinaryR + NN-AryR$ NbinaryR, NN-AryR
Measurement instrument Decision criteria Metric	GenMETRIC tool See Table AIV See Table AII	Uses	Measurement function Indirect metric	—	
Direct metric Indirect metric Indicator	See Table AII See Table AIII See Table AIV	Uses Satisfies	Analysis model Decision criteria Indicator Information need	See Table AIV IME, IC, IH, INM, CI To know the E/R diagram’s maintainability	

ACKNOWLEDGEMENTS

This research is partially supported by the MAS project ‘Dirección General de Investigación of the Ministerio de Ciencia y Tecnología’ (TIC 2003-02737-C02-02) and by the MECENAS project ‘Consejería de Ciencia y Tecnología, Junta de Comunidades de Castilla-La Mancha’. We would also like to thank professionals of Cronos Iberica S.A. for their active participation and collaboration and anonymous reviewers for their comments and suggestions that have helped us to improve this work.

REFERENCES

1. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Software Engineering Institute: Carnegie Mellon University, Pittsburgh, PA, 1995.
2. *Capability Maturity Model Integration (CMMI), version 1.1*. Software Engineering Institute: Carnegie Mellon University, Pittsburgh, PA, 2002.
3. *Software Process Assessment—Part 7: Guide for Use in Process Improvement*. ISO IEC 15504 TR2:1998, International Organization for Standardization: Geneva, 1998.
4. *Quality Management Systems—Fundamentals and Vocabulary*. ISO/IEC 9000:2000, International Organization for Standardization: Geneva, 2000.

5. Florac WA, Carleton AD. *Measuring the Software Process. Statistical Process Control for Software Process Improvement*. Addison-Wesley: Reading, MA, 1999.
6. Software Process Engineering Metamodel (SPEM), adopted specification. Object Management Group, 2002.
7. Avison D, Lan F, Myers M, Nielsen A. Action research. *Communications of the ACM* 1999; **42**(1):94–97.
8. Polo M, Piattini M, Ruiz F. Using a qualitative research method for building a software maintenance methodology. *Software Practice and Experience* 2002; **32**(13):1239–1260.
9. García F, Ruiz F, Piattini M. Definition and empirical validation of metrics for software process models. *Proceedings of the 5th International Conference Product Focused Software Process Improvement (PROFES'2004)*, Kansai Science City, Japan (*Lecture Notes in Computer Science*, vol. 3009). Springer: Berlin, 2004; 146–158.
10. García F. FMESP: Marco de Trabajo Integrado para el Modelado y la Medición de los Procesos Software. *PhD Thesis*, University of Castilla-La Mancha, Spain, 2004.
11. García F et al. An ontology for software measurement. *Technical Report UCLM DIAB-04-02-2*, Computer Science Department, University of Castilla-La Mancha, Spain, 2004. Available at: <http://www.info-ab.uclm.es/trep.php?&codtrep=DIAB-04-02-2>.
12. XML Metadata Interchange (XMI) specification, version 1.2. Object Management Group, 2002.
13. Meta Object Facility (MOF) specification, version 1.4. Object Management Group, 2002.
14. Ruiz F, Piattini M, García F, Polo M. An XMI-based repository for software process meta-modeling. *Proceedings of the Product Focused Software Process Improvement (PROFES'2002)*, Rovaniemi, Finland (*Lecture Notes in Computer Science*, vol. 2559). Springer: Berlin, 2002; 546–558.
15. García F, Ruiz F, Piattini M, Polo M. *Conceptual Architecture for the Assessment and Improvement of Software Maintenance (Enterprise Information Systems IV)*, Piattini M, Filipe J (eds.). Kluwer Academic: Dordrecht, 2002; 219–226.
16. García F, Ruiz F, Cruz JA, Piattini M. Integrated measurement for the evaluation and improvement of software processes. *Proceedings of the 9th European Workshop on Software Process Technology (EWSPT'9)*, Helsinki, Finland (*Lecture Notes in Computer Science*, vol. 2786). Springer: Berlin, 2003; 94–111.
17. METRICA version 3.0. *Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información*. Ministerio de Administraciones Públicas, Spain, 2000. Available at: <http://www.csi.map.es/csi/metrica3/>.
18. Genero M, Jiménez L, Piattini M. Measuring the quality of entity relationship diagrams. *Proceedings of the 19th International Conference on Conceptual Modeling (ER 2000)*, Salt Lake City, UT (*Lecture Notes in Computer Science*, vol. 1920), Laender AHF, Liddle SW, Storey VC (eds.). Springer: Berlin, 2000; 513–526.
19. Calero C, Piattini M, Genero M. Empirical validation of referential integrity metrics. *Information Software and Technology. Special Issue on Controlled Experiments in Software Technology* 2001; **43**(15):949–957.
20. *Software Engineering—Software Measurement Process*. ISO/IEC 15939, International Organization for Standardization: Geneva, 2002.
21. García F, Bertoa M, Calero C, Vallecillo A, Ruiz F, Piattini M, Genero M. Towards a consistent terminology for software measurement. *Information and Software Technology* (submitted).
22. Jones C. Making measurement work. *CROSSTALK The Journal of Defense Software Engineering* 2003; **16**(1):15–19.
23. Paulk MC. A comparison of ISO 9001 and the capability maturity model for software. *Technical Report CMU/SEI-94-TR-12*, Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, PA, 1994.
24. Mutafelija B, Stromberg H. *Systematic Process Improvement Using ISO 9001:2000 and CMMI*. Artech House: Norwood, MA, 2003.
25. Goldenson D, Jarzombek J, Rout T. Measurement and analysis in capability maturity model integration models and software process improvement. *CROSSTALK The Journal of Defense Software Engineering* 2003; **6**(7):20–24.
26. Piattini M, Genero M, Jiménez L. A metric-based approach for predicting conceptual data models maintainability. *International Journal of Software Engineering and Knowledge Engineering* 2001; **11**(6):703–729.
27. Genero M, Piattini M, Calero C (eds.). *Metrics for Software Conceptual Models*. Imperial College Press: London, 2005.
28. Manso E, Genero M, Piattini M. No-redundant metrics for UML class diagrams structural complexity. *Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003)*, Klagenfurt/Velden, Austria, 2003; 127–142.
29. Serrano M, Calero C, Trujillo JC, Luján-Mora S, Piattini M. Empirical validation of metrics for conceptual models of data warehouses. *Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004)*, Riga, Latvia (*Lecture Notes in Computer Science*, vol. 3084). Springer: Berlin, 2004; 506–520.
30. Aurum A, Jeffery R, Wohlin C, Handzic M. *Managing Software Engineering Knowledge*. Springer: Berlin, 2003.
31. Vizcaino A, Favela J, Piattini M, García F. Supporting software maintenance in Web repositories through a multi-agent system. *Proceedings of the Atlantic Web Intelligence Conference (AWIC'03)*, Madrid, Spain (*Lecture Notes in Computer Science*, vol. 2663). Springer: Berlin, 2003; 307–317.
32. Bezivin J, Breton E. Applying the basic principles of model engineering to the field of process engineering. *UPGRADE: European Journal for the Informatics Professional* 2004; **V**(5):27–33. Available at: <http://www.upgrade-cepis.org/issues/2004/5/upgrade-vol-V-5.html>.