# Proposed Data Model for a Historical Base Tool

Karine Santos Valença, Edna Dias Canedo, Ricardo Ajax D. Kosloski, and Sérgio A. A de Freitas
Faculdade UnB Gama - FGA
University of Brasília (UnB)
Caixa Postal 8114 – 72405-610
Brasília, DF, Brazil
E-mail: valenca.karine@gmail.com, ednacanedo@unb.br, ricardoajax@unb.br, sergiofreitas@unb.br

**Abstract - Measurement processes, specifically in Software Engineering, are of great relevance to the success of an organization, though relevant data can only be obtained from systematic measurements. In this case, it is important to note that to attain proper results, the necessary inputs for collecting measures must be collected by the organisation itself. Therefore, organisations should have a measurement historical database that should be kept updated. In this paper we present a proposal where the users can define their metric using a historical database system built on the basis of metamodelling, so that they can create their metric, registering this metric in the system and, if necessary, modifying the definition of the metric without making changes to the application's source code. Furthermore, based on this metamodel, a data model was developed to support the application's software development. So, the use of our metamodel to build an organisational historical database software system means to boost the creation of new metrics, or to update the definitions of the metric already in existence and, in this way, improve the users' agility and flexibility to maintain the organisation's metric. The improvement in this aspect will reflect on the capability of the organisation to take advantage of the results from its measurements processes.**

***Keywords - Measurement; Systematic Mapping; Metamodelling; Goal Question Metric; Database.***

## I. INTRODUCTION

Measurement processes are fundamental for many types of organisations as regards their knowing, controlling, and streamlining their productive processes [6]. The Software Engineering domain is no different, as measurement processes allow the teams to understand its capabilities and thus allowing the planning and execution of solid software projects with respect to its costs, scopes, quality, risk and other variables [7]. Measurement processes cover a wide range of elements, amongst which the sub processes that define, collect, analyse, and report the results to the whole organisation and that are very important to support many different kinds of actions [6], [22].

In order to improve the organisation's agility and efficiency in the use of their measurement data, it is important that the majority of these sub processes is done in the most automated possible way. Moreover, the speed of their acquisition and precision of the measurement records will be vital to allow the continuous and critical analysis of comparative studies (benchmarking), as they are fundamental to support decisions about corrective and preventive actions, as well as for improvement opportunities in the organisation itself [14]. Beyond this, the actions and the measurable results of the organisation should be continuously evaluated to show that such actions and measurable results are always aligned with the improvement goals of the organisation [2], [22].

Based on corrective and preventive actions, the organisations should adjust their metric, which can be a big problem for their historical database systems if the organisation were to do huge maintenance work in the code of already-implemented applications.

A possible solution to boost the speed and efficiency in the updating of the organisation's historical database would be to deal with metamodelling resources.

In this kind of solution, the users could define the features of the metric adopted by the organisation without necessarily proceeding with code maintenance to run the organisation's historical database.

In this paper, the solution proposed has as one advantage the possibility to generate several measurements that comply with a given standard assumed by the organisation, by just defining or updating the definition of its metric.

This work aims at, as a general goal, using the concepts of metamodelling as a practical application to improve the maintainability of the software development organisation's historical database systems.

This paper is structured as follows: Section II presents the method for research used in this study, as well as the research questions that were defined. Section III presents a view of the results found until now, and Section IV provides the conclusions and our expectations for future work.

## II. RESEARCH METHOD

Research was done as a systematic mapping literature review theory, to define research questions, find relevant publications, select the publications, and extract the data related to the research questions [17].

Questions related to the subject were proposed and a search strategy was created to answer them. The search strategy was the one defined by the Experimental Software Engineering area, to build a reliable knowledge base from the use of query strings applied to academic publications databases such as the ACM, IEEE, amongst others [20]. The publications that met the search criteria were selected to be part of the study.

### *A. Research Questions*

Research questions were raised that were aligned with the goal of this work, to build a solid knowledge base that would lead to a metamodelling proposal for a measurement system. The measurement is the cornerstone of an experimental study,

ISBN: 978-1-61208-498-5

being defined as the mapping of the experimental world towards the formal or relational domain. The main goal of the mapping is to characterise and manipulate the attributes of empirical entities in a formal way. Instead of directly judging, based on real entities, the numbers or symbols are assigned to such entities, and the appraisal is done based on such numbers and symbols. The number or symbol attributed to the entity via the mapping is named measurement. The attribute of the entity under measurement is named metric [20].

The questions prepared were:

(**QP1**) What does the literature characterise as metamodelling?

(**QP2**) What is a measurement and what is its relevance in Software Engineering?

(**QP3**) Which metamodelling resources can be used to build a flexible measurement historical base?

(**QP4**) What data model can be used to create a flexible historical base tool?

## B. *Search Strategy*

The search strategy included a manual search of publications found in the Computer Science and Software Engineering areas. For each item, and using related key-words, the research sources were the ACM Digital Library, IEEE Xplore, SpringerLink, and Science Direct.

The key words were picked according to the survey questions. At first, key words with a wider coverage were defined, to find the largest number possible of related publications. More precise, key words were defined during the execution of the survey, to answer some specific questions. Table 1 shows the key words used in the search and the number of papers found in each academic database.

Along with the publications found, we also discovered new research sources, using the snowball technique, to find research objects. This works as a chain where a research object provides other research sources which, in turn, spawn other sources, and so on [21]. As an example, paper [13] holds a simplistic definition of what the Goal Question Metric (GQM) method is and the reasons to use it. A reference to that section leads to publication [2], a paper with over 400 mentions, according to Google Scholar.

TABLE I. NUMBER OF PAPERS FOUND FOR EACH KEY WORD

| Key word | ACM | IEEE Xplore | Springer Link | Science Direct |
|---|---|---|---|---|
| Metamodelling | 697 | 539 | 2,088 | 1,189 |
| Metamodelling software | 466 | 275 | 1,722 | 802 |
| GQM Method | 11 | 41 | 936 | 519 |
| Measurement metamodel | 38 | 63 | 1,436 | 1,676 |
| Goal Question Metric Metamodel | 0 | 1 | 614 | 557 |

## C. *Criteria for Inclusion and Exclusion*

After implementing and executing the search strategy, publications were included that had a title or abstract that referred to the theme of our research and published between 1989 and 2016. Year 1989 was chosen as a starting point since the oldest information of relevance to the theme occurred that year.

The exclusion criterion filtered out the papers with fewer than 10 mentions according to Google Scholar. As it is expected, the papers that were not widely referenced might not have the expected quality, or have a superficial approach of the subject.

With the goal of screening the publications, we read the abstract of the publications finally listed and verified whether they addressed some of the points raised earlier. From then on, 28 publications had been listed which, after a more detailed reading and analysis were cut down to 19 publications.

# III. RESULTS

This section describes the results obtained after the systematic mapping of the selected papers.

## A. *Analysis of the First Research Question*

(**QP1**) What does the literature characterise as metamodelling?

To answer this, a characterisation of metamodelling was done to then answer what metamodelling is. The following results were obtained.

Mellor, Clark, and Futagami define model as a grouping of components that describe physical and abstract things, or a hypothetical reality [16]. Rothenberg [18] characterises models as an abstraction from reality as models cannot represent all of its aspects. Thus, we can characterise models as a group of elements aimed at expressing/simplifying a given reality.

Models cannot represent reality with all of its particularities and, moreover, one could say that models do not need to represent all the aspects of reality. In creating a model, one can consider only the points that are relevant for the proposal and, as a result, using models allows dealing with reality in a more simplistic way, reducing its complexity and irreversibility [18].

Another important point of this simplicity relates to the visual representation of the model. In [15] it is said that ‘each model will be expressed using a combination of text and multiple complementary and interrelated diagrams’. As a result, with the reduction of the aspects that would be represented, the model is visually cleaner and more organised.

To begin understanding what metamodelling is, it is interesting to look at the origin of the word. The word ‘metamodel’ is a variant of ‘model’ and thus it is understood that metamodelling is a specific kind of modelling [9]. Metamodelling is the act of producing metamodels. The word meta, according to the Oxford Dictionary, is of Greek origin and means ‘behind’ or ‘after’. In the meaning that we seek, a metamodel means that which is behind a model.

In [14] a metamodel is defined as a model for a modelling language. For example, a model for the Unified Modelling Language (UML) is described by an UML metamodel which defines how models can be structured, as well as the elements they may contain [15]. The work presented by Clark [3] also states that a metamodel describes a modelling language with a higher level of abstraction than that of the modelling language [3]. To simplify these definitions we may conclude that a metamodel defines how a model should be constructed.

A metamodel is also considered as a model, although the metamodel captures the main features and properties of the language that will be modelled and, apart from that, a metamodel has its own architecture, named meta-metamodel which defines how metamodels should be described [3]. Thus, it is understood that the difference between a common model and a metamodel is that the information represented by a metamodel is actually a model [9].

The use of metamodelling has its benefits. It allows the definition of languages without the need for implementing technologies, focusing on the domain of the problem and promoting an increase of productivity throughout the development process [3].

### *B. Analysis of the Second Research Question*

(**QP2**) What is a measurement and what is its relevance in Software Engineering?

As presented in [6] a measurement is 'the process through which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules'. This means that, through measurement, it is possible to describe something, through observation and recording, assigning numerical values to its characteristics.

The work presented Kan [11] states that a 'measurement is crucial to the progress of all sciences. Scientific progress is made through observations and generalizations based on data and measurements…' It is no different in Software Engineering, and measurements have a very important role to play in the success of organisations, as such measurement processes allow the software teams come to grips with their capacities. As a result, with measurements it is possible to carry out solid project planning that will not overshoot the planning done as regards the scope, quality, risk, and project length, as the measurement allows one to gain knowledge on the processes [7]. Apart from that, there are many characteristics in software projects that can be measured [12], strengthening the importance of measurement in this area.

Measurement in the domain of Software Engineering corresponds to the successive process of defining, collecting, and analysing data in the software development process, to understand and control the processes [19]. That is, through measurement, several useful items of information are studied and analysed and, through them, one can discover how the process is executed, what results are being generated in it, and also learn about managing the process, making the process better.

In the domain of measurement, and within Software Engineering, an approach is used, named Goal Question Metric (GQM) to define the measurements. GQM is an approach based on the premise that the measurements should be defined based on the measurement goals of the organisation, which in their turn generate questions which, again in their turn, can be answered via metric. Apart from that, the structure of the GQM provides a framework to interpret the measured data, using the established metric, and their associations with the questions put forward and the results measured, which serve as inputs to meet the measurement goals. [2].

The GQM method consists of defining goals and then refining them into questions aimed at characterising the object of the measurement, and supporting the interpretation of the data on the goals [2], [19]. It should be pointed that, for the questions to support interpretation in a satisfactory manner, the research points should be set on an intermediary level of abstraction [19], that is, neither being so specific, as regards the metric, nor abstract as regards the goals.

For each question, it is suggested that the measurement team creates hypotheses for them, and such hypotheses will be compared with the results obtained during the interpretation stage [19]. Defining the hypotheses is important as it serves to gauge the level of knowledge of the team on the processes executed.

The hypotheses are then refined into metric which is data aimed at answering the questions in a quantitative manner. The metric can either be objective or subjective [2].

The definition of the metric in the GQM is done on a top-down direction, meaning that at first a general perspective of the measurement is defined (the goal of the measurement) and after that it is processed into finer detail until the entire specification is reduced to base elements (metric) [5]. The method has 4 stages [19]:

**Planning stage**: Selection of the project to apply the measurement to; project definition, characterisation, and planning.

**Definition stage**: Definition and documenting of the goals, questions, metric, and hypotheses.

**Data collection stage**: Collection of the present data for the measurement project.

**Interpretation stage**: Analysis of the data as related to the set metric, answering to the questions raised; after that the evaluation takes place - whether the goal been reached.

An important activity takes place during the data collection stage. In it, a definition is made as to how the data will be collected, and how they will be filled in/entered, and how the data will be stored in a database [19]. It is important to store this data as, with the analysis of such historical data one will be able to identify patterns and also plan more adequately, as well as improve the processes in the organisation [7].

### *C. Analysis of the Third Research Question*

(**QP3**) Which metamodelling resources can be used to build a flexible measurement historical base?

The metamodelling language requires a specific metamodelling architecture. The traditional metamodelling architecture has 04 meta-levels [3]:

**M3**: holds the meta-model that describes the properties all metamodels can display.

**M2**: holds the metamodel that captures the language.

**M1**: hold the application, and can contain the classes of an object-oriented system or the table for a relational database.

**M0**: holds the data of the application developed.

In using this architecture, the authors of work [8] propose a conceptual framework for measurement. Level M3 holds an abstract language for metamodel definition. Level M2 holds generic metamodels that will serve to create specific models. This level consists of the measurement metamodel and the domain metamodels that work to represent the types of entities eligible for measurement. Level M1 is that where the specific models are and it consists of the measurement and domain models. And Level M0 holds the data the users record in the applications [8].

Apart from the use of the architecture for the construction of a metamodel, one should define the abstract syntax, after defining the syntactic rules and the meta-operations, and define the concrete syntax, and define the semantics and, lastly, establish the relation with other modelling languages [3].

A metamodel should describe the concrete and abstract syntax, as well as the semantics of the model. The concrete syntax relates to the notation that facilitates the presentation and construction of the models, and may be text-based or visual-based. The abstract syntax describes the definition of concepts, the relationships that exist between them, and how they blend to create models [18]. Semantics gives meaning to each concept, in a well-defined and well-set manner [10]. The merger of these three factors allows the construction of a coherent and well-structured metamodel.

After the study on how to build a metamodel, a study was done of metamodels that had already been put forward by other authors on measurement. Figure 1 shows the measurement metamodel, using the GQM method, developed based on [4], [8], [19].

According to the GQM method, it is expected that a measurement project should have several goals, if it intends to measure in an intentional manner [2].

A measurement goal has a standard structure, a template, as proposed by Basili [2]; this proposal states that a goal has the object to be measured, the characteristic of the object, the standpoint, and the proposal for measurement. For example, the goal of ‘Maximising client satisfaction with the software product’, has the ‘software product’ as an object, ‘satisfaction’ as a characteristic, the ‘client’ as a standpoint, and ‘maximising’ as the proposal.

This template was finely tuned in the work of [19] that proposes the following structure: analysing (the object of measurement), for a proposal of (the goal of the measurement), as regards the (focus on object quality), from the standpoint of (those that measure the object), and in the domain of (the realm where the measurement will be made).

Using this template allows greater compliance with the GQM method as it is more aligned with its proposals and definitions. Based on these templates, Table 2 was built with the structure of the goal, with the GQM method [2], [19].

Following the definition of the goal, it will be refined into several questions, aimed at characterising the object that will be measured. The questions, on their turn, are answered by a metric that aims at answering with quantitative information [2].

A metric can consist of several measures and the latter defines the structure for the measurement values [4]. To define this structure it is necessary to choose the measurement unit (table, column, percentage, hours, etc.), the scale to be used (integers, real, etc.), and the type of scale (Scale types are nominal, ordinal, interval, ratio, and absolute).

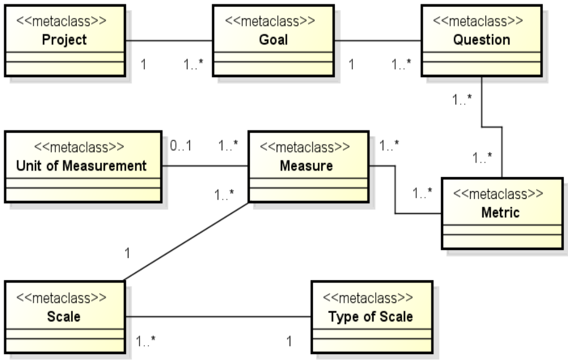

Figure 1. Metamodel proposed for measurement, using the Goal Question Metric method

TABLE II. GOAL STRUCTURE AS PROPOSED BY THE GQM METHOD

| Analyse | Goal of measurement |
|---|---|
| **For the proposal of** | The goal of the measurement (understanding, improving, or controlling) |
| **As regards** | The focus on the quality of the object to be measured |
| **From the standpoint of** | The people that measure the object |
| **In the domain of** | The environment where the measurement takes place |

Scale types are nominal, ordinal, interval, ratio, and absolute. In the nominal scale the categories for attributes should be jointly exhaustive (all categories as a whole should cover the possibilities of the category of an attribute) and mutually excluding (the attributes may be classified in only one category). In an ordinal scale the attributes may be compared amongst them in an ordinal manner, but this scale does not provide information on the magnitude of the difference that exists between points of the scale. The interval scale allows knowing the difference between two points of the scale. This type of scale requires a well-defined measurement unit that can be considered a standard and that is repeatable. The ratio scale is similar to the interval one, as in the proportional scale it is possible to find the difference between two points of the scale. However, in the proportion scale it is possible to find an absolute zero and non-arbitrary [11]. The absolute scale is merely a count of some element in the entity and it allows any arithmetical analysis [6]. Most of the measurements are done in the interval scale and that of the arbitrary proportion [11].

Having created the measurements, one should define how the measurements relate to each other to create a metric. This means defining the function of measurement for that metric. This is an important stage as it is from this function that the result of the metric is obtained.

To better explain metamodelling, the following example was created: The project of a given software company has, as one of its measurement goal, to ‘Analyse the software product to improve its quality from the standpoint of the clients’. Faced with this, the clients were asked a set of questions. One of these questions was: ‘What is the quality of the software product developed by the company?’ To answer it, a metric was raised, namely ‘average errors per class’. This metric has two dimensions: the numbers of errors found and the number of classes in the project. The unit for measurement ‘number of errors found’ is the quantity of Errors and, for measurement ‘number of classes’ it is the quantity of Classes. The scale for both measurements is of integers from 0 to infinite and the type of scale is the absolute scale.

Figure 2 was created to better illustrate what information the user should provide to define a metric in the proposed measurement system. In this figure the example is about one measurement called Average error per class.

One metric holds many measures. The measures blend through mathematical operators or functions to define a metric, that is, a metric can be obtained through a mathematical combination of its dimensions. It should be pointed that the metric ‘Average errors per class’ is obtained with the combination of two independent variables, ‘Number of Errors’ and ‘Number of Classes’. This combination is done with the use of the ‘average’ function, which is a resultant from a mathematical formula that blends the independent variables to calculate the dependent variable, i.e., ‘average errors per class’ would be the ‘number of errors’ <divided by> ‘number of classes’. What the users need to do to define the ‘average errors per class’ metric is to blend the two measures, using this operator. Each measure will have a measurement unit and also a scale. In the example given above, both measures have the same scale, although they can be different, even when belonging to the same metric. Having defined the metric, it is necessary to choose the type of scale.

Another example, if the metric Productivity = Size/Effort, then size and effort are the basic measurements (independent variables) used to calculate productivity. Thus, the productivity metric is obtained with the blending of these measurements and the <divided by> operator. They have a scale, a measurement unit, and a type of scale.

In order for a user to define one's metric one should first record the measurements (independent variables), use the mathematical operators or functions that were previously defined in the application and blend them as the ‘calculation formula’ for the metric (dependent variable).

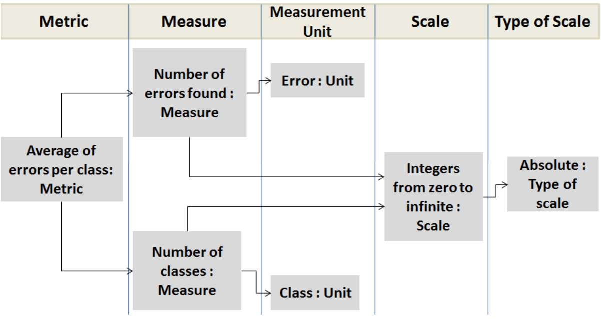

Figure 2. Definition of metric using the proposed metamodel

### D. Analysis of the Fourth Research Question

(**QP4**) What data model can be used to create a flexible historical base tool?

To answer this, we thought about what data would be required for a user to enter in order to carry out measurements with the use of the GQM method. And as a result the data form below was created:

**Project – defined by:**
Project name; Project description;
Company name
And Start date:
**Measurement Goal (as described by GQM Method):**
Analyze:
For a proposal to:
As regards:
From the standpoint of:
In the domain of:
**Questions – defined by:**
Question; Description of question:
**Metric – defined by:**
Name of metric;
Description of metric;
Date of calculation.
**Measurement function:**
Measurement 1:
Measurement 2;
Measurement no.:
Another value:
Function:
**Measurement:**
Name of measurement:
Description of measurement:
Date of collection:
Value:
**Measurement Unit:**
Name of measurement unit:
Description of measurement unit:
**Scale:**
Name of scale:
Number set:
Minimum number:
Maximum number:
**Type of scale:**
Name of type of scale:
Description of type of scale:

Based on this data form it was possible to create a data model. The data model created went through a verification procedure and a validation walk-through, the latter being an informal technique to ascertain software product quality [1]. As a result, after the creation of the model, those who took part in the walk-through did an analysis of the model to try and find inconsistencies in it. Those who took part in the verification and validation procedure are students of the Software Engineering School at the Gama UnB School, enrolled in and attending the Software Verification and Validation classes. This discipline has the following pre-requisites for students to attend: Metric and Estimates for

Software, Software Quality, and Advanced Software Development. Apart from that the lecturer of this course followed the process, to give his feedback regarding the model proposed.

Following the analysis work the model was explained to the participants, to solve eventual queries. After that the items the participants considered wrong were discussed and a consensus was reached on the model. The model was re-structured based on these discussions and a check list was produced to verify whether the model had been corrected in its entirety.

The data model derived from the metamodel proposed as it has all the metaclasses of the metamodel, apart from including the attributes of such classes and other tables for standardisation the model.

## IV. CONCLUSION

Based on the study presented, it is possible to see the contribution of measurements to software development companies, allowing people to learn about the performance and quality of their processes and products developed. Thus, when using metamodelling in a measurement system, it is expected that an increase in productivity and in measurement activities will take place. Apart from that, metamodelling also grants additional flexibility in the definition of metric as it allows users to create their own metric, no longer being dependent of existing ones.

With the proposal made in this work it will also be possible to create an application that allows users, apart from creating their own metric, to store such measurement data in a historical data base and, due to the flexibility metamodelling allows, it will be possible to make several changes to those metric without the need to do maintenance on the source code.

As a result, in the continuation of this work, the data model proposed in this paper will be used to create an application that will allow users to define their own metric and record them in a historical data base. The development process in its initial stage and we expect that several organisations might benefit from it, cutting their costs with system maintenance, especially in legacy systems.

## REFERENCES

[1] O. Balci, Validation, verification, and testing techniques throughout the life cycle of a simulation study. Proceedings of Operations Research, pp. 121-173, 1994.

[2] V. R. Basili, Software Modelling and Measurement: The Goal Question Metric Paradigm. Computer Science Technical Report Series, CS-TR-2956 (UMIACS-TR-92-96), University of Maryland, College Park, MD.1992.

[3] T. Clark, P. Sammut, J. Willans, Applied metamodelling: A foundation for language driven development, 2nd Ed. Ceteva, 2008.

[4] E. Damiani, A. Colombo, F. Frati, and C. Bellettini, A Metamodel for Modelling and Measuring Scrum Development Processes. LNCS, pp. 74-83, 2007.

[5] B. El-Haik, and A. Shaout, Software design for six sigma. Wiley. 2010.

[6] N. Fenton, and S. Pfleeger, Software Metric. Boston: PWS Pub., 1997.

[7] W. A. Florac, R. E. Park, and A. D. Carleton, Practical Software Measurement: Measuring for Process Management and Improvement Pittsburgh, PA, 1997.

[8] F. García, M. Serrano, J. Cruz-Lemus, F. Ruiz, and M. Piattini, Managing software process measurement: A metamodel-based approach. Information Sciences, pp. 177-182, 2007.

[9] P. C. González, and B. Henderson-Sellers, Metamodelling for software engineering. John Wiley. 2008.

[10] D. Harel, and B. Rumpe, Meaningful modelling: what's the semantics of "semantics"? Computer. 37, 10, pp. 64-72, 2004.

[11] S. Kan, Metric and models in software quality engineering. Addison-Wesley. 1995.

[12] L. Laird, and M. Brennan, Software measurement and estimation. Wiley-Interscience. 2006.

[13] L. Lavazza, Providing automated support for the GQM measurement process. IEEE Softw. 17, 3, pp. 56-62, 2000.

[14] C. Lokan, T. Wright, P. Hill, and M. Stringer, Organisational benchmarking using the ISBSG Data Repository. IEEE Softw. 18, 5, pp. 26-32, 2001.

[15] S. Mellor, MDA distilled. Addison-Wesley. 2004.

[16] S. Mellor, A. Clark, and T. Futagami, Model-driven development - Guest editor's introduction. IEEE Software, pp. 14-18, 2003.

[17] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, Systematic mapping studies in software engineering, Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), British Computer Society, Swinton, UK, pp. 68-77, 2008.

[18] J. Rothenberg, Artificial intelligence, simulation & modelling. Chapter: The Nature of Modelling, John Wiley &Sons, Inc., New York, NY, USA, pp. 75-92, 1989.

[19] R. Solingen, and E. Berghout, The goal/question/metric method. The McGraw-Hill Companies. 1999.

[20] E. A. G Amaral, G. H. Travassos, "Em Busca de uma Abordagem para Empacotamento de Experimentos em Engenharia de Software". In. Proceedings of the 2nd JIISIS - Jornada Ibero-Americana de Engenharia de Software e Engenharia de Conhecimento. Salvador, Brazil. 2002.

[21] W. Vogt, Dictionary of statistics and methodology. Sage Publications. 1993.

[22] ISO/IEC 15939:2007 - Systems and software engineering — Measurement process, International Organisation for Standardization, Geneva, Switzerland, 2007.