JISTEM: Journal of Information Systems and Technology Management
E-ISSN: 1807-1775
tecsi@usp.br
Universidade de São Paulo
Brasil

Souza Neto, João; Nunes Ferreira Neto, Arthur

METAMODEL OF THE IT GOVERNANCE FRAMEWORK COBIT

JISTEM: Journal of Information Systems and Technology Management, vol. 10, núm. 3, septiembre-diciembre, 2013, pp. 521-540
Universidade de São Paulo
São Paulo, Brasil

JISTEM - Journal of Information Systems and Technology Management
Revista de Gestão da Tecnologia e Sistemas de Informação
Vol. 10, No. 3, Sept/Dec., 2013 pp.521-540
ISSN online: 1807-1775
DOI: 10.4301/S1807-17752013000300004

METAMODEL OF THE IT GOVERNANCE FRAMEWORK COBIT

João Souza Neto
Arthur Nunes Ferreira Neto
Catholic University of Brasilia, Brasília/DF, Brazil

ABSTRACT

This paper addresses the generation and analysis of the COBIT 4.1 ontological metamodel of IT Governance framework. The ontological metamodels represent the logical structures and fundamental semantics of framework models and constitute adequate tools for the analysis, adaptation, comparison and integration of IT best practice frameworks. The MetaFrame methodology used for the construction of the COBIT metamodel is based on the discipline of conceptual metamodeling and on the extended Entity/Relationship methodology. It has an iterative process of construction of the metamodel's components, using techniques of modeling and documentation of information systems. In the COBIT 4.1metamodel, the central entity type is the IT Process. The entity type of IT Domain represents the four domains that group one or more IT processes of the COBIT 4.1. In turn, these domains are divided into one or more Activities that are carried through by one or more Roles which are consulted, informed, accounted for or liable for each Activity. The COBIT 4.1 metamodel may suggest adaptation or implementation of a new process within the framework or even contribute to the integration of frameworks, when, after the processes of analysis and comparison, there are connection points between the components and the logical structures of its relationships.

Keywords: COBIT, Metamodels, Entity/Relationship, IT Governance, IT framework.

Manuscript first received/*Recebido em* 23/01/2012 Manuscript accepted/*Aprovado em*: 01/07/2013

Address for correspondence / Endereço para correspondência

João Souza Neto, Doctor of Science in Electrical Engineering, University of Brasilia – UNB, Professor at Catholic University of Brasilia, on the Master's degree Program in Information Technology and Knowledge Management, Campus Avançado, SGAN 916 Asa Norte - Modulo B - Sala A121 - CEP:70.790-160 Brasília – DF, Brasil - Telefone: (61) 3448-6534 - E-mail: joaon@ucb.br.

Arthur Nunes Ferreira Neto, Master in Information Technology and Knowledge Management – MGCGI/UCB, Catholic University of Brasilia, Researcher at Catholic University of Brasilia in Metamodels of IT frameworks, Campus Avançado, SGAN 916 Asa Norte - Modulo B - Sala A111 - CEP: 70.790-160 Brasília – DF, Brasil - Telefone: (61) 3338-6534 - E-mail:arthurnetobsb@gmail.com.

1. INTRODUCTION

According to the IT Governance Institute (2005), "the survival and success of an organization on the new global market, where time and distances were suppressed, depend on the effective management of information and related technologies." In this context, where IT (Information Technology) plays a decisive and strategic role within the organizations, models of IT best practices frameworks have emerged in the last two decades. These frameworks are a response of business owners to the challenges posed by IT governance and management, working as tools for the promotion of the alignment between the IT processes and the strategic objectives of the organization.

According to Johannsen and Goeken (2007), the IT best practices frameworks "describe organizational objectives, processes and aspects of the IT management and control of IT".

The effective implementation of an IT best practice framework is a complex activity that demands planning and managing and it usually induces significant changes in the organization and in its processes. Thence, the challenge arises to deeply understand the structure of the framework so that a preliminary study of its suitability to the organization's processes can be made.

Furthermore, it has been observed that the adoption of only one of these IT best practice frameworks may not be sufficient for a particular organization. Despite the different foci and the conceptual and structural differences, IT best practice frameworks, in principle, are not incompatible, and they can be used concomitantly to promote an improvement in the organization’s IT management. Therefore, one of the challenges currently faced in IT management is how to analyze, adapt, compare, and integrate different IT best practice frameworks.

Consequently, it is understood that the first step towards solving these problems is understanding the logical structures and the generating semantics of the IT best practice frameworks. This can be achieved through the methodical generation of ontological metamodels (models of models) of these frameworks.

The basis for this proposition is that the ontological metamodels represent, from a higher level of abstraction, the conceptual components and the rich logical structure and semantics of the relationships of the IT best practices frameworks and, at the same time, they enable the adaptation, comparison and integration among different IT frameworks.

Among the main approaches used, up to now, in order to carry out the analysis and comparison of IT best practice frameworks, there are the high-level classifications based on diverse criteria of comparison and the high-level detailed mapping of the functions and processes among the frameworks (ITGI, 2006, 2008).

However, only the application of these two approaches does not significantly contribute to the solution of comparing the IT best practice frameworks problem. The high-level classifications based on comparison criteria are not detailed enough to detect correspondences or incoherencies among different areas of the IT frameworks. On the other hand, the detailed mapping of the functions and processes of the IT best practice frameworks shows a high level of detail, but it presents little available information for understanding the conceptual and logical structures which are important for the planning and the effectiveness of the integration.

This paper, in an effort to fill this gap, used the MetaFrame methodology, which comprises procedures, strategies and instructions for creating ontological-type metamodels for these IT best practice frameworks. (Ferreira Neto, 2010). This methodology is then applied for the generation of the COBIT 4.1 framework.

2. THEORETICAL REFERENCE

2.1 Metamodels Definitions

The managing of elements in an organization increasingly uses more and more complex models, tools, and environments of modeling. For Karagiannis (2002), the state of the art in the area of organizational modeling is based on metamodels.

A literal analysis about the meaning of a metamodel may start with the prefix “meta”. In Greek, “meta” means “that which is beyond”, “that which encompasses”, “that which supersedes”, “that which transcends”, etc.

According to the open consortium of the OMG (Object Management Group), responsible for the MDA (OMG, 2003) and UML (OMG, 2004) specifications, a model is an instance of a metamodel, which implies that a metamodel is a model of another model.

An important contribution to the studies which were developed concerning the subject of this paper was provided by Atkinson and Kühne (2003a and 2003b), who identified two dimensions of metamodeling that generated two distinct forms of instancing of the metamodel objects (linguistic and ontological). One dimension is related to the definition of the language and it uses the linguistic instantiation, employed, for example, in MDA architecture, the basis of UML language. Another dimension concerns the definition of the domain or type of object and uses the ontological instancing employed in the creation of the metamodel of the COBIT framework in this study. Both forms occur simultaneously and serve to precisely locate an element of the model in the linguistic-ontological space.

Figure 1 uses the OMG-MDA architecture with four layers of abstraction (M_0 to M_3), also followed by UML2.0 and MOF 2.0 linguistic modeling standards. There is the visualization of a linguistic metamodel with four horizontal layers that starts with M_0, denoting the lowest level, and M_3, the highest level of abstraction. At the same time, there is the visualization of the ontological metamodel, represented by different areas separated by a dashed line in the vertical division at the M_1 level. By expliciting the two metadimensions, Figure 1 also illustrates the relationship between the elements of the model and the real world. The dog and the lamp (mental concept) of the M_0 level are the elements of the real world to be modeled. The real Lassie is “represented” by the object Lassie and not by an ‘instance of’ Collie. The abstraction level M_1 contains the first level of abstraction of an object in the real world, together with the type of which the object is an ontological instantiation. The Lassie object (O_0) is an ontological instantiation of the type Collie (O_1). From M_1 each level is a model expressed in the language defined at the higher level. In M_2, the Lassie object is a linguistic instantiation of the Object type, which, in $M_{3,}$ is a linguistic instance of the Class type.

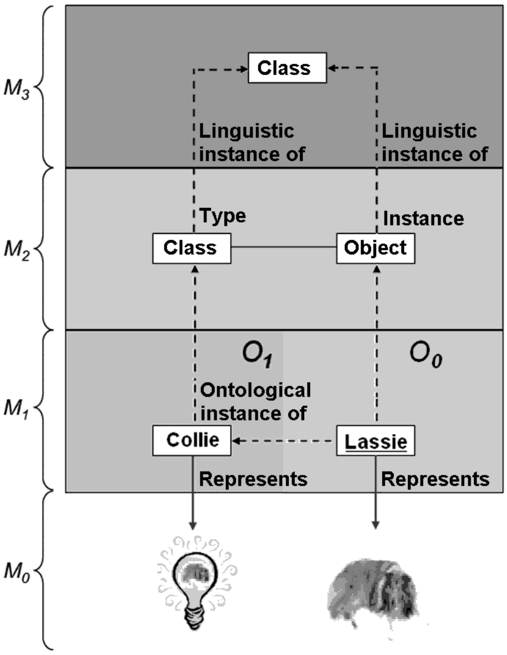


Figure 1: The Linguistic Metamodel (Adapted from Atkinson and Kühne, 2003b)

The ontological metamodels employ the 'instance of' relationship to relate the concepts to their types or metatypes. In Figure 2, the ontological levels were extended by rotating Figure 1 to the right, and adding level O_2.Therefore, the ontological metalevels are arranged horizontally. For Atkinson and Kühne (2003b), the two points of view are equally valid and useful.

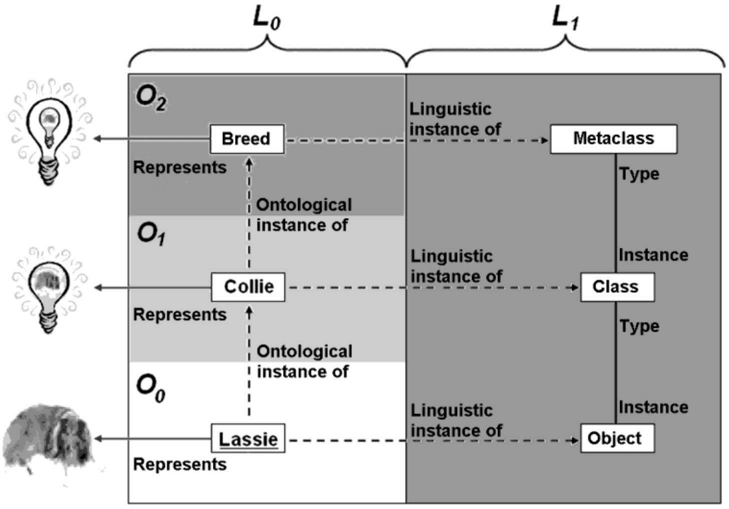


Figure 2: The Ontological Metamodel (Adapted from Atkinson and 2003b).

According to Atkinson and Kuhne (2003b), despite the validity and utility of the ontological metamodels of types, the tool builders and members of the standardizing consortia, such as the OMG, the metamodel term refers typically only to the metamodel of the linguistic type. Meanwhile, from the perspective of the user of the language, the hierarchy of types formed by ontological levels is much more relevant. In other words, the ontological metamodels are metamodels for the users focused on the content and the linguistic metamodels are a standard of metamodels focused on forms.

Researcher Strahringer (1996) studied how the level hierarchies of the models are built and coined the term ‘metaization principle’ to designate an operation that is repeatedly applied from a level to another, or rather, the primary mechanism of abstraction to structure the objects in levels of hierarchy. Kühne’s analysis (2006) is similar to Strahringer’s (1996), but it uses a different distribution of the elements for the levels and a diverse terminology. The MetaFrame methodology (Ferreira Neto, 2010) utilizes the metaization principle in order to verify and inform users how the metamodel components of the COBIT framework were built.

The most used metaization principle in information systems is the linguistic metamodeling. For instance, the syntax of the languages of modeling is at the M_2 level, such as the well-known E/R (Entity/Relationship) methodology by Chen (1976) that is applied to represent part of the objects in the real world (M_0) at the level of an E/R (M_1) model, where only the components of the language (types, entity, relationship types, attributes etc.) can be used. Based on this principle, a M_2 level structures the representation of the objects at the M_0 level in the M_1 level. In the ontological metamodeling, metatypes at the M_x level are defined and they describe the concepts that exist at the M_{x-1} level.

2.2 Metamodels Principles and Instructions

Schütte (1998) is one of the authors who contribute to this research work through the modeling instructions contained in GoM (Guidelines of Modelling). The GoM is a framework for the development and evaluation of conceptual models composed of six general principals, described as follows:

1. Construction Adequacy Principle: There must exist a consensus among specialists and users on what type of a model construction is adequate for the problem and its proposal.

2. Language Adequacy Principle: the language used to create the metamodel fulfills its proposal. This principle refers to the completeness and the consistency between the model and the metamodel. This means that the model should not possess any symbol or item that has not been specified in the metamodel.

3. Economic Efficiency Principle: this principle formulates economic restrictions on the task of modeling. The costs of developing of a model should not surpass the gains of its use.

4. Clarity Principle: this principle deals with the comprehensibility and expressivity of the model. Within the objectives of clarity, there are the hierarchical decompositions, the formatting (arrangement of the elements) of the model and the filtering of information. Criteria and objectives of the quality of the graphic formatting of a model were defined by Tamassia (1988).

5. Systematic Conception Principle: this principle deals with the consistency of the construction among the models and it is also important for the integration of the models.

6. Comparability Principle: this principle deals with the semantic comparison between two models according to their correspondence or similarity. This is one of the most important principles in a metamodelling environment. Metamodels are frequently used to compare and integrate models.

Goeken (2009) proposes the use of the principles defined by Schütte (1998) to also evaluate the metamodels. The author adds three new specific instructions to evaluate the quality of the metamodels:

Instruction 1: a metamodel reveals its metaization principle. It is important for the metamodel user to know which rules were utilized to construct the metamodel levels.

Instruction 2: a metamodel should possess a clear mapping between the universe of the discourse and the words and symbols that name and describe them. There should not exist doubts among users concerning the meaning of concepts in the metamodel.

Instruction 3: a metamodel must have rich semantic connections. The relationships among the metamodel components must be relevant and described in an expressive way.

The metamodels created from the MetaFrame methodology should be verified concerning the principles and instructions described.

2.3 Applications of the Metamodels

The ontological metamodels can be applied in order to complete the analysis, adaptation, comparison and integration of the IT Governance frameworks. Once the components of the metamodels are extracted, the frameworks can be examined and analyzed so that the characteristics of their structure are known. This analysis contributes to the evaluation of the framework and also in helping the implementation and adaptation within the organization.

Other possibilities related to the application of the IT Governance frameworks metamodels are the comparison and integration with different frameworks. Using the same methodology for the construction or, according to Strahinger (1996), the same metaization principle, the representation of the metamodels allows the comparison between the frameworks at a higher or abstract level. This comparison process is an important step towards the integration of the frameworks. The integration of the metamodels can guide the integration of the frameworks at a lower or concrete level.

2.4 Extensive E/R Methodology

The Entity Relationship E/R methodology, proposed by Chen (1976), was developed for the creation of conceptual and semantic models. The metamodels constructed with the MetaFrame methodology, presented in this study, follow the concepts and the notation of an extension of the E/R methodology, formalized by Engel et all (1992), with the objective of improving metamodel expressiveness. Figure 4 presents the main components and their notation, according to the authors cited above.

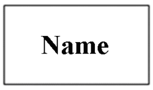
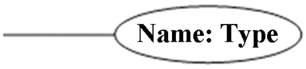
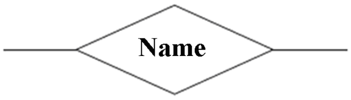
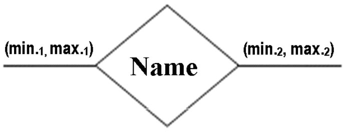
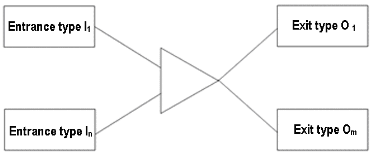
Object	Visual Representation
Entity type: classes of objects (concrete or abstract) with the same properties, which have their own meaning and which can be distinctly identified.	
Attribute: describe the properties of characteristics of an entity type. They receive a value of a particular type of data.	
Relationship type: facts, the occurrences that link two or more entity type objects. The type, relationship, is a common grouping form of various entity types.	
Cardinals: integrity restrictions. Each entity type participant that is a relationship type can be restricted by the minimum and maximum cardinal numbers (min. max.) An entity type can participate in at least minimum, min., and maximum, max., relationships with another entity type. The symbol o "*" is used in place of the numbers to denote an indeterminate number of associations.	
Constructor type: classification of the entity types. Starting with one or more entity types, entry, one or more types of exit can be obtained. They are used to represent a generalization, specialization, and partitioning of entity types.	 Entrance type I1 Entrance type In Exit type O 1 Exit type Om

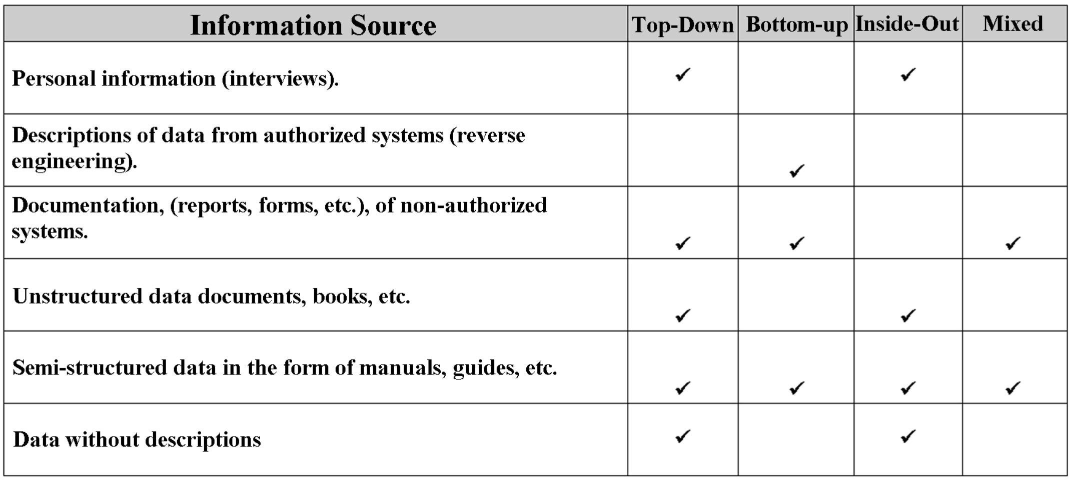
Figure 3: Components and notation of the extended E/R methodology. Adapted from Engels et all (1992).

Depending on the quantity and complexity of the objects (entity types, relationship types, attributes, and constructor types), the use of a modeling strategy is important to help in the organization and development of the work of finding and defining the metamodel components. One modeling strategy for the extended E/R methodology is a sequence of steps that repeat themselves, producing small transformations of the initial model in the final model. The choice of the strategy for the construction of the model is influenced by the main source of information of the modeling process.

The literature shows that there are four types of basic modeling strategies (Top-Down, Bottom-Up, Inside-Out or Middle-Out and Mixed). However, there is no consensus among the authors on which of these is the best technique. The works of Heuser (1998) and Atzeniet (1999) are used to describe these strategies. In the Top-Down strategy, an initial model is created in which the most abstract concepts ('from above') are represented first. Afterwards, intermediary models are gradually created through the refinement of the concepts into more specific concepts.

The Bottom-Up strategy (from below to above) is the inverse of the Top-Down strategy (from above to below). It consists in starting with the most elementary and detailed concepts to construct more abstract and complex concepts. The Inside-Out strategy (from inside to out) or Middle-Out strategy (from the middle out) consists in starting with the considered most important or central concepts (from inside), and then gradually adding peripheral concepts related to them (to outside). The Mixed strategy is a combination of the other strategies.

None of the modeling strategies presented above is universally accepted. The authors recommend the use of a certain strategy or a combination of them, starting with the specific information. Figure 5 shows some sources of information and recommendations on strategies to be used.



Information Source	Top-Down	Bottom-up	Inside-Out	Mixed
Personal information (interviews).	✓		✓	
Descriptions of data from authorized systems (reverse engineering).		✓		
Documentation, (reports, forms, etc.), of non-authorized systems.	✓	✓		✓
Unstructured data documents, books, etc.	✓		✓	
Semi-structured data in the form of manuals, guides, etc.	✓	✓	✓	✓
Data without descriptions	✓		✓	

Figura 4: Modeling strategies by source of information. Source: the authors.

The complexity of the model depends on the types of sources of information and on the quantity of the entity types to be represented. Therefore, in more complex models, with more than 20 types of entities, various strategies are usually used at the same time. In these cases, a higher level model is divided so that each partition can be modeled separately.

2.5 The COBIT 4.1 Framework

The COBIT 4.1 (Control Objectives for Information and related Technology) is a guide for IT management and governance, organized to ensure that the use of IT resources are effectively aligned with the organization’s business strategies. According to ITGI, the COBIT's mission is "to research, develop, publish and promote a control framework for the governance of Information Technology that is updated and internationally accepted for adoption by organizations and is used in a day-to-day basis by business managers, IT professionals and auditors" (ITGI, 2007). It is probably the most widely used reference framework for IT governance (SIMONSSON e JOHNSON, 2006a), risk mitigation and value delivering through IT (RIDLEY et al, 2004; DEBRACENY, 2006).

The conceptual model of COBIT 4.1 is represented by a cube whose faces are interrelated, as shown in Figure 5.

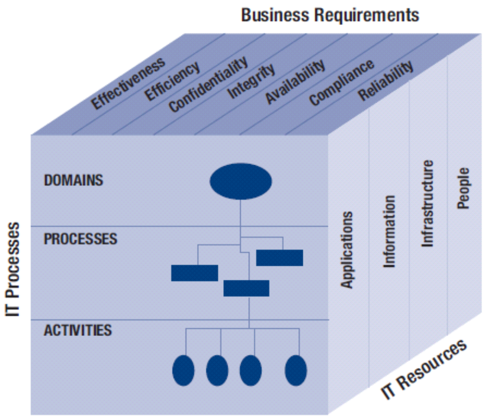


Figure 5 –The Cube of COBIT 4.1. Source: ITGI, 2007.

To better understand the model, the IT Process dimension is organized in a structure with four domains, as follows: Planning and Organization – it focuses on strategy and tactics so that IT may actually contribute to the business goals of the organization; Acquisition and Implementation - the focus is on the implementation of the IT strategy. In this domain the solutions are identified, developed, acquired, implemented and integrated with business processes; Delivery and Support –focusing on issues related to the delivery of services, including routine operations, security, continuity and training; and finally Monitoring and Evaluation - its goal is to regularly assess the IT processes from a quality and compliance point of view according to control requirements.

These four domains include thirty-four processes and these processes comprise two hundred and ten activities.

On the other side of the cube, there are Business Requirements. According to the model proposed by COBIT 4.1, in order to satisfy business objectives, information needs to conform to certain criteria such as effectiveness, efficiency, confidentiality, integrity, availability, compliance, and reliability.

Finally, the third dimension links characteristics related to the IT resources, which are: Applications, Information, Infrastructure and People, to previous dimensions. The areas of focus for the IT governance, according to the COBIT 4.1, are presented in the pentagon illustration shown in Figure 6 (ITGI, 2007).

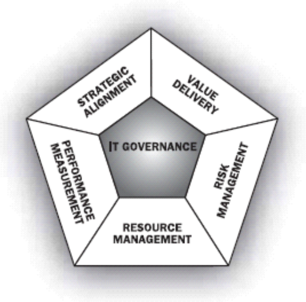


Figura 6 – IT Governance Areas of Focus. Source: (ITGI, 2007)

At the Pentagon, one can identify the strategic alignment, which aims to ensure consistency between the organization's strategic goals and the IT objectives; the value delivery, which is linked to the delivery of products or services with appropriate quality, time and cost that allows to achieve the objectives previously agreed upon; the risk management, which refers to the treatment of uncertainties and to the value preservation; the resource management, which aims to ensure the capacity to support the activities required by the business, optimizing costs and other available resources, and, finally, the monitoring of the performance of IT activities with the purpose of ensuring the management of the entire environment.

To meet managerial control and measurement IT’s needs the COBIT 4.1 provides guidelines for the thirty-four IT processes which contain assessment and measurement tools for the IT environment of the organization including maturity model, critical success factors, key goal indicators and key performance indicators for each process (GREMBERGEN, 2004).

3. RESEARCH METHOD

The survey, according to Gil (2002), is a "formal and systematic development of the scientific method. The fundamental objective of the research is to find answers to problems by employing scientific procedures." Moresi (2004, p.30) adds that "research is a reflective and critical procedure for seeking answers to problems not yet solved."

The research is classified according to the research methodology that will be employed. In this work it was used used the classification of Vergara (2000), for whom the research can be classified according to its purposes or goals and the means of research or technical procedures.

Regarding its purposes or goals, this research is classified as a methodological and applied research. The research methodology is the study related to the development of instruments to capture or manipulate reality. Therefore, it is associated with paths, shapes, manners, and procedures used to reach a determined purpose. The research is applied to solve specific problems, more immediate or not. Therefore, it has a practical purpose, unlike pure research that is motivated primarily by the intellectual curiosity of the researcher and is set mainly at the speculation level.

As to the means of research or to the technical procedures, this research is classified as bibliographical. The bibliographical research may be defined as the development of a systematic study based on materials published in books, articles/papers, periodicals, electronic networks or, in other words, material that is accessible to the general public. Although it provides analytical tools for any type of research, it can also be an end in itself. The published material may come from a primary or secondary source. Table 1 summarizes the classification of this research.

Table 1: Classification of this research according to Vergara (2000).

Categories	Tipos	Justificativa
Purposes or goals	Methodological	Development of a methodology for building meta-frameworks of IT best practices (MetaFrame).
	Applied	Several practical applications of the methodology and results of the research in organizations and professional applications: metamodels creation, analysis, adaptation, comparison and integration of frameworks of IT best practices.
Means of investigation or technical procedures	Bibliographical	Search of the best methodologies, strategies and guidelines for the creation of the methodology for this research. Use of the frameworks official guides for gathering and analyzing data.

4. METHODOLOGY

In order to develop the COBIT 4.1 metamodel, a collection, depuration, organization, analysis and presentation of data was made. The ITGI's three official COBIT 4.1 guides were used as sources of information (*http://www.isaca.org/Knowledge-Center/COBIT*). The process of data collection of official documents is similar to the data survey technique of systems analysis for the modeling of information systems. The Extended Entity/Relationship methodology, by Engels et all (1992), was used, combined with the conceptual modeling strategies for the organization and analysis and representation of the data according to the following types: entity type, relationship type, attribute type and constructor type. The final purpose of this data survey was to develop the conceptual metamodeling framework.

All of the procedures described above are included in the methodology named MetaFrame, which was created by Ferreira Neto (2010), and it describes a detailed process of creation and verification of the quality of the metamodels of IT best practices. The objective of the MetaFrame methodology is to ensure the quality of the metamodel and create useful products such as metamodel data dictionaries to be used in the applications of metamodels, as, for example, in the comparison and integration of frameworks.

4.1 The MetaFrame Methodology

The aim of this methodology is to create a metamodel framework of IT best practices based on the collecting and analyzing of data contained in the official guides of the IT best practices framework. The methodology comprises an iterative construction process of the metamodel components using modeling techniques and documentation of information systems, thus determining the verification of the results based on quality criteria.

The metamodel documentation, generated by the MetaFrame methodology, is important for the analysis, adaptation, comparison and integration of the IT frameworks as it contains a data dictionary with the definitions of the components represented.

Phase 1 of the Metaframe methodology comprises the preparation of the study. In this phase, the objectives are defined, the professionals are selected and their roles are assigned and the training and the distribution of support materials for the participants are performed. Phase 2 is the execution phase, where the metamodel data collection and the iterative construction and documentation processes of the metamodel are carried out using modeling techniques. Phase 3 verifies the quality of the metamodel according to the principles and instructions presented in the 2.2 item as well as the correction and updating of the documentation generated by the methodology. A summary of the methodology is presented in Figure 7.

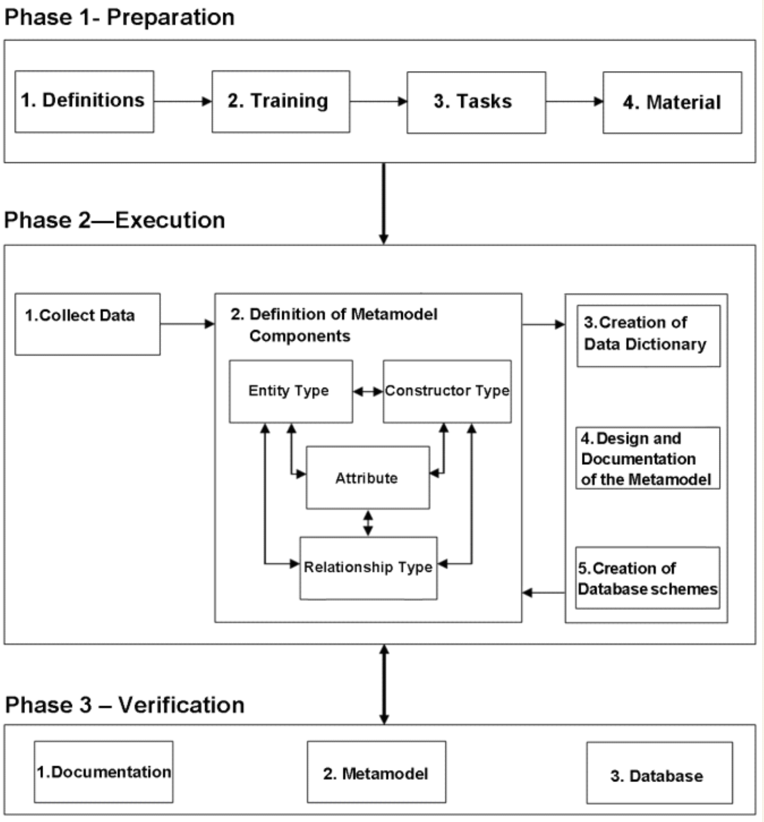


Figure 7: Metaframe Methodology for the creation of IT metamodel frameworks.

At the end of the verification phase of the MetaFrame methodology, the results or products will be ready to be disclosed within the organization or published outside of it. The metamodel and the explanatory summary should be released together so that the users will have no questions as to the components represented. After the release of the metamodel, the team responsible for its development may receive questions from the users, as well as suggestions for the improvement and the implementation of the metamodel. It is suggested that the members of the team that developed the metamodel work meet to analyze the issues and suggestions from the users and to take the necessary actions. It is also important that the team discuss what was learned from the creation of the metamodel, based on the MetaFrame methodology.

5. RESULTS AND DISCUSSIONS

5.1. Building the COBIT 4.1 Metamodel

The steps to build the ontological metamodel of COBIT 4.1, according to the Metaframe methodology are presented in this item. The beginning of the job to create a metamodel corresponds to Phase 1 of the MetaFrame methodology, named Preparation. This phase consists of the following stages: the reading the ISACA's official guides for the COBIT 4.1such as the COBIT 4.1 Manual, the COBIT 4.1 Control Practices and IT Assurance Guide, Using COBIT 4.1; the ontological metamodel creation; the metadata dictionary creation;, the database schemas of the metamodel creation and the analysis and customizing of the model.

The job of creating the COBIT 4.1 metamodel itself corresponds to Phase 2 of the MetaFrame methodology, named Execution, where the following activities are performed: the data collection, the definition of the metamodel components, the creation of the data dictionary, the creation of the metamodel and the creation of database schemas.

In Phase 3 of the MetaFrame methodology, named Verification, the products that are generated, are then examined as to their correctness and quality. The following stages are performed in this phase: the documentation, the metamodel and the database.

The diagram of the COBIT 4.1 metamodel is shown in Figure 8.

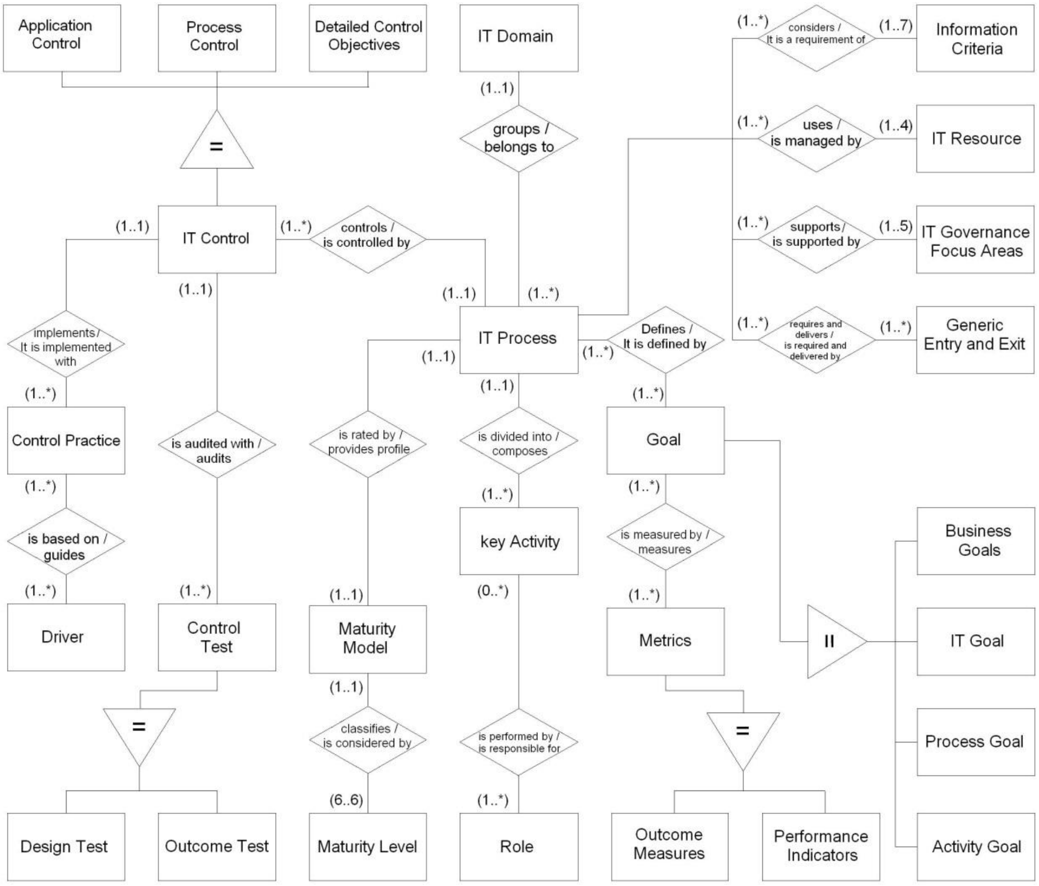


Figure 8: COBIT 4.1 Metamodel. Source: The authors.

5.2. Summary of the COBIT Metamodel

The ontological metamodel developed in this work represents the conceptual structures that constitute the COBIT 4.1 framework. These concepts are symbolized as entity types (rectangles), relationship types (diamonds), cardinalities (numbers in parentheses), attributes (ellipses), constructor types (triangles) and lines connecting entities to relationships. The explanatory summary, recommended by the MetaFrame methodology in phase 2, stage 4, step 2, intends to give a clear interpretation of the metamodel to the user. The definitions presented here are a selection from the ISACA's COBIT 4.1official guides.

In the COBIT 4.1 metamodel, the central entity type is the IT Process. The COBIT 4.1 has thirty-four IT processes that belong to certain domains of IT. The IT Domain entity type represents the four domains that group one or more IT processes of the COBIT 4.1.

The COBIT 4.1 processes are divided into one or more Activities. Each Activity of COBIT 4.1 is carried out by one or more Roles that are consulted, informed, accountable or liable for each Activity.

Each COBIT 4.1 IT Process considers from one up to seven elements of the Information Criterion entity type as business requirements for information. Each IT Process also uses from one up to four elements of the IT Resource entity type (applications, people, information and infrastructure).

An IT Process also supports from one up to five elements of the IT Governance Focus Area entity type. Each IT Process requires and delivers one or more elements of the Input and Output entity type, containing results (documents, actions, etc.) of the COBIT 4.1 IT processes or of external processes.

Each COBIT 4.1 IT Process is evaluated according to a specific Maturity Model. The Maturity Model entity type provides a maturity profile for each process based on a rating of just six elements of the Maturity Level entity type.

An IT Process defines one or more elements of the Goal entity type. A Goal entity type of the COBIT 4.1 represents the following entity types: Business Goal, IT Goal, Process Goal or Activity Goal. One Goal is measured by one or more elements of the Metric entity type. A Metric in the COBIT 4.1 represents the Key Goal Indicator or Key Performance Indicator entity types.

A COBIT 4.1 IT Process is controlled by one or more elements of the IT Control entity type. The IT Control entity type represents the following entity types: Application Control, Process Control or Detailed Control Objective and it controls a particular IT Process.

An IT Control is implemented with one or more elements of the Control Practice entity type. The Control Practice implements only a particular IT Control. A Control Practice is based on one or more elements of the Driver entity type (risk or value).

An IT Control is audited by one or more elements of the Control Test entity type which, in turn, audits a specific IT Control. The Control Test entity type represents the Design Test and Result Test entity types.

5.3. Relevant Issues of the COBIT 4.1 Metamodel

During the metamodel creation process some issues have arisen, and they were addressed to in meetings held to discuss and review both the data collection stage and the process of analyzing and defining the components of the metamodel.

The COBIT 4.1 metamodel uses four constructor types that aim to optimize the representation of the entity types as well as the relationship types involved. They were formed through the specialization of an input entity type into two or more output entity types. This structure allowed the use of only one relationship type with the input entity type making it easier to understand the metamodel. Another advantage of using constructor types is the decomposition of a more general concept into more detailed concepts which are important for the understanding of the metamodel.

An example of the constructor type use was the creation of the IT Control entity type and its specialization in the output entity types: Application Control, Process Control and Control Objective. The three output entity types are examples of controls for IT processes of the COBIT 4.1. They come up together in COBIT's Control Practices and IT Assurance Guide Using COBIT that are related to the control practices and the audit tests, respectively.

Another issue of the construction of the COBIT 4.1 metamodel was the creation of the Business Goal entity type. This type of goal is not present in the IT processes forms in COBIT 4.1 User's Guide, but the goals of activity, process and IT are. As the MetaFrame methodology involves thorough reading of all the official guides for data collection of the metamodel components, significant references to the Business Goal entity type were found externally to the forms of IT processes. In Appendix I of the Handbook of COBIT 4.1, there are tables containing the seventeen business goals suggested by ISACA and related to IT processes.

The COBIT 4.1 presents a RACI chart for each IT process containing IT activities and the type of role (R-Responsible, A-Accountable, C-Consulted, I-Informed) for each process stakeholder. The metamodel represents the type of relationship "is performed by" between an Activity entity type and a Role entity type. The responsibility of each role was considered as an attribute of the relationship because it depends on the elements of the two entity types at the same time. The responsibility is dependent of the Activity and Role entity types as shown in Figure 9. The attribute responsibility, as well as other attributes of the metamodel entity types, appears in the diagram of the complete metamodel, printed on a page large enough to hold all the components. However, in cases where the attribute is indeed important for the comprehension of the metamodel by the users, a given atribute may be included in the metamodel drawing.

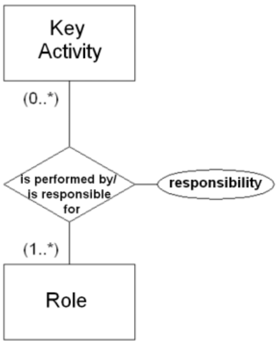


Figure 9: Representation of the Relationship Attribute

One issue that sparked the debate among the participants of the meeting was the creation of the Input and Output entity type. The COBIT 4.1 provides for each IT process one table with the entries for the process, including the external inputs and another for the results or outputs of the process. Both inputs and outputs of the processes can be in the form of actions, documents, etc. Instead of creating two entity types, one for the input and another for the output, it was decided to create a single entity type as the elements of the entity type are of the same type.

In order that the metamodel demonstrates the two types of relationship of the Input entity type and the Output entity type with the IT Process entity type, two verbs were used for naming the relationship type. One verb expresses the relationship with the inputs of the process and another verb expresses the relationship with the outputs of the process. Therefore, it is clear to the user of the metamodel what the scope of the relationship between the two entity types is. Reading the metamodel from the left to the right, the relationship is described as "requires and delivers", i.e., an IT process requires one or more inputs and also delivers one or more outputs of the Input entity type and the Output entity type.

Another important issue discussed by the participants was the representation of the maturity model of the COBIT 4.1. The COBIT 4.1 framework has a maturity model for each IT process. This model was based on the maturity levels of the CMM (Capability Maturity Model) developed by the SEI (Software Engineering Institute), although it has different goals. The maturity model of the COBIT 4.1 is not meant to accurately assess the level of the process maturity. The maturity model of the COBIT 4.1 fosters the creation of a maturity profile for the IT process by evaluating the process evolution stage with each of the six levels of the maturity model.

5.4. Validation of the COBIT 4.1 Metamodel

Table 2 shows the validation of the COBIT 4.1 metamodel based on the principles defined by Schütte (1998) for evaluating metamodels plus the guidelines defined by Goeken (2009). This table also shows that the generated metamodel fully meets all quality requirements set by these two authors.

Table 2: Validation of the COBIT 4.1 Metamodel

Principles(P) / Guidelines(G)	Fulfillment
P1 - Adequacy of construction	The use of ontological metamodels to portray the COBIT 4.1 essential concepts and structures was adequated.
P2 - Adequacy of language	The purpose of the metamodel is suitable for the language used for the COBIT 4.1 metamodel does not include other symbols or different items from those already present in the model, i.e., there is consistency between the model and the metamodel.
P3 - Economic efficiency	The ontological COBIT 4.1 metamodel sustains this principle because it does not need any extension or modification in the language used, therefore it does not require an additional development for the organization.
P4 – Clarity	This principle is fulfilled by the guidelines and principles used in the metamodel creation, for example, the quality goals and criteria of the graphical formatting.

P5 – Systematic Design	This principle deals with the construction consistency between the metamodels. This principle is covered by the systematic creation of the metamodels of IT best practices frameworks through the use of the MetaFrame methodology.
P6 – Comparability	This principle handles the semantic comparison between two models according to their similarity or correspondence. This principle is met because the metamodels created using the MetaFrame methodology are comparable as they have in their documentation a metamodel data dictionary that allows a more effective comparison of the concepts presented.
O1 - Revelation of the Metaization Principle	This guideline is attained through the revelation that the type of the COBIT 4.1 metamodel is the ontological one and that the language components of the extended E/R methodology (rectangles - entity type, lozenges - relationship type etc.), used to represent the metamodel, are named according to the essential or primary concepts of the model, i.e., according to the concepts that classify the model.
O2 – Clear Mapping	The metamodel of the COBIT 4.1 meets this principle because the MetaFrame methodology demands the creation of an explanatory summary of the metamodel so the users may clearly understand the concepts used. Although those concepts are known to the users of the model, they may not be known to other professionals of the IT management area
O3 - Having rich semantic connections	This guideline is followed by the COBIT 4.1 metamodel, which is created according to the MetaFrame methodology because only the most significant relationships that express important concepts contained or created from the model are used. For every relationship there are two expressions naming the type of each relationship, depending on the direction in which the entity types are read. The names of the relationships are extracted from the official guides, with rare exceptions.

6. CONCLUSION AND FUTURE RESEARCH

This paper aims at presenting the COBIT 4.1 metamodel and it was used to analyze the overall structure of the framework.

Comparing the treatment followed here with other approaches on the subject, such as that defined by Goeken (2009), it is important to emphasize that the development of the COBIT 4.1 metamodel was started from the so called MetaFrame methodology, and not from the use of a simple conceptual modeling devoid of standard procedures that prevent the repetition of the modeling by others. Furthermore, it is worth noting the fact that the MetaFrame methodology requires strict adherence to the official documentation of the model under study. In other approaches, it remains clear the differencesbetween the conceptualization of the official documents and what is described by the author.

In future research, it is also possible to use metamodels as a methodological support for adapting or customizing frameworks to the processes and structures of an organization. For example, the COBIT 4.1 metamodel may suggest how to adapt or implement a new process into the framework by displaying the associated entity types and relationship types. This means that the metamodel, having a rich conceptual framework and their relationships, becomes a guide for the changes to be made in the framework, since its essential characteristics are respected.

One may also compare the IT best practices frameworks using metamodels, which can be quite useful for analyzing possible feature additions. This can be done using the documents generated by the MetaFrame methodology, in particular, the metamodel data dictionary with all its components and their descriptions, types, relationships etc. The metamodel dictionary generated by the MetaFrame methodology is a prerequisite for comparing the structures of two or more frameworks and, also, for addressing the issue of synonyms and homonyms concepts.

The metamodels of the IT best practices frameworks can also contribute to the integration of frameworks. Here, the term integration is used in the context of creating a common area between two frameworks, despite keeping, at the same time, the characteristics of each one. After the processes of analysis and comparison, the connection points between components and the logical structures of their relationships are identified. Then, it is possible to construct a new metamodel displaying an integration area containing the components of the frameworks that were integrated. Entity types such as processes, activities, resources, products etc., are present in most of the IT best practice frameworks with similar meanings and attributes. Other components, despite having different names, have the same meaning and can also be integrated.

The metamodels may also contribute to the fusion of different IT frameworks. The term fusion refers to the creation of a new framework starting from the existing ones. This situation occurs when it is requested that one framework complements another, what may occur through the embodiment of the more specific framework into the structure of the framework displaying a wider scope. The fusion process also depends on the analysis and comparison of the frameworks. After this step, a new framework will arise from the introduction or adaptation of existing concepts or by creating new concepts consistent with the frameworks of origin.

REFERENCES

Atkinson, C., and Kühne, T. (2003a), Model-Driven Development: A Metamodeling Foundation, IEEE Software, vol. 20,no. 5, pp. 36-41.

Atkinson, C., and Kühne, T. (2003b), Calling a Spade a Spade in the MDA Infrastructure, International Workshop Metamodeling for MDA, York.

Atzeni, P., Ceri, S., Paraboschi, S., Torlone, R. (1999), Database Systems Concepts, Languages and Architectures. McGraw-Hill.

Chen, Peter, P.S. (1976), The Entity-Relationship Model: Towards a Unified View of Data, ACM Transaction on Database Systems, vol. 1, nº1, pp. 9-36.

Debreceny, R. Re-engineering IT Internal Controls: Applying Capability Maturity Models to the Evaluations of IT Controls. Proceeding of the 39nd International Conference on Systems Engineering Research, Hawaii, 2006.

Engels, G., Gogolla, M.,Hohenstein, U., Hulsmann, K. (1992), Conceptual Modelling of Database Applications Using an Extended ER Model. North Holland, Amsterdam, pp. 157-204.

Ferreira Neto, A. N. F., Metamodelos Ontológicos de Frameworks de Melhores Práticas de TI (2010). Disponível em:

http://www.bdtd.ucb.br/tede/tde_busca/arquivo.php?codArquivo=1282. Acesso em: 8mar 2011.

Gil, Antônio C. Como elaborar projetos de pesquisa. 4ª Ed. São Paulo: Editora Atlas, 2002.

Goeken M., Alter S. (2009), Towards Conceptual Metamodeling of IT Governance Frameworks Approach - Use - Benefits, hicss, 42nd Hawaii International Conference on System Sciences, pp.1-10.

Grembergen, W. Strategies for information technology governance. Idea Group Inc., 2004.

Heuser, C. A. (1998), Projeto de Banco de Dados, 6ª edição. ISBN: 979-85-7780-382-8. Editora Bookman.

IT Governance Institute. COBIT 4.1 (2005). Disponível em: http://www.isaca.org/Knowledge-Center/COBIT/Pages/Downloads.aspx. Acesso em: 1 Fev 2011.

IT Governance Institute: COBIT Mapping: Overview of International IT Guidance (2006), 2nd Edition, ISBN 1-933284-31-5.

IT Governance Institute: COBIT® 4.1. Framework Control Objectives Management Guidelines Maturity Models (2007).

IT Governance Institute COBIT® Mapping: Mapping of ITIL V3 With COBIT® 4.1 (2008), ISBN 1-933284-31-5.

Johannsen, W., Goeken, M. (2007). Referenzmodelle für IT Governance. verlag GmbH, Heidelberg.

Karagiannis, D., and Kühn, H. (2002), Metamodeling Platforms. In A. Min Tjoa, & G. Quirchmayer (Eds.), Lecture Notes in Computer Science: Vol. 2455. Proceedings of the Third International Conference EC-Web, Springer, pp. 451-464.

Kühne, T. (2006), Matters of (Meta-) Modelling, In Journal on Software and Systems Modeling, Volume 5, Number 4, pp. 369-385.

Moresi, E. A. D. Metodologia de Pesquisa. Brasília-DF: Universidade Católica de Brasília-UCB, mar. 2004.

OMG (2003). MDA Guide Version 1.0.1 Version 1.0.1, OMG document omg/03-06-01.

OMG (2004).UML-Unified Modeling Language Infrastructure Specification, Version 2.0, Version 2.0, OMG document ptc/03-09-15.

Ridley, G.; Yo Ung, J. e Carroll, P. COBIT and Its Utilization: A Framework from the Literature. Proceedings of the 37th International Conference on System Sciences, Hawaii 2004.

Schütte, R., Rotthowe, T. (1998). The Guidelines of Modeling- an approach to enhance the quality in information models. In Ling, Ram, Lee (Eds.) Conceptual Modeling – ER 98. Singapore, 16.-19.11.98, pp. 240-254.

Simonsson, M. e Johnson, P. Defining IT Governance – A consolidation of Literature. Department of Industrial Information and Control Systems. Royal Institute of Technology (KTR), 2006.Disponível em <http://www.ics.kth.se/Publikationer/Working%20Papers/EARP-WP-2005-MS-04.pdf>. Acesso em 05/02/09.

Strahringer, S. (1996).Metamodellierungals Instrument des Methodenvergleichs, Shaker Verlag, Aachen.

Tamassia, D.; DI Battisti, G.; Batini, C.(1988): Automatic graph drawing and readability of Diagrams. IEEE Transactions on Systems, Man and Cybernetics, 18 1, S. 61-79.

Vergara, Silvia Constant, (2000) Projetos e Relatórios de Pesquisa em Administração. 3ª ed. São Paulo: Atlas.