Abstract - Up to now, there is little academic support for the challenges of IT management. As a reaction, various best practice frameworks were developed, which can be subsumed under the topic IT governance. These still have no sound basis or scientific foundation. To undertake a step in this direction we present a metamodel of COBIT, the popular IT governance framework. A major goal of this paper is to represent the underlying logical and semantically rich structure of this framework. This turns out to be fruitful for comparing and integrating different frameworks. Furthermore, frameworks can be checked for completeness and can be integrated on this basis. An interesting application is the representation of IT governance frameworks in a tool, which is demonstrated in the paper as well.

Keywords: Frameworks, IT Governance, Metamodelling

1. Introduction

While there is little guidance for the management of IT in general and for specific challenges like business/IT alignment or risk management in particular [3], [4], the majority of computer science research deals with system development and related issues. To offer guidance for management tasks and governance challenges seems critical because in enterprises, usually a higher percentage of expenditure is spent on 'running IT' rather than systems engineering and development of new systems.

Due to the fact that there is a clear need for methodological support for current tasks and challenges of IT management and IT governance, it is surprising, that little attention is paid to these questions. [6] and [17] censured researchers for the lack of effort put into evaluating e.g. how business and IT can be properly aligned, how IT related risks can be managed and how IT can contribute to the overall value of the enterprise.

In recent years, there were some associations and public institutions like ISACA (Information Systems Audit and Control Association) and CCTA (Central Computer and Telecommunication Agency) /OGC (British Office of Government Commerce) that developed frameworks (e.g. COBIT and ITIL) to support management and governance of IT. These frameworks are well established in practice [10], [15]. However, there is a lack of theoretical foundation, from a scientific viewpoint.

This paper undertakes steps towards the theoretical foundation of best practice frameworks by proposing to model them as metamodels.

In the following section (II) we discuss some definitions of IT governance which helps to reveal different viewpoints on the subject. In section III we discuss the concepts ‘model’ and ‘metamodel’ as well as their relationship. After that we present a metamodel of COBIT, the popular IT governance framework of the ISACA (section IV). Furthermore, we discuss the advantages and the application of this metamodel and show some research in progress – e.g. a prototype with which we represent the COBIT framework in an application.

2. IT Governance

IT Governance is a relatively new field of research. So it is not surprising that the current definitions of IT governance show many differences. In the following part we compare three widespread definitions to explain the different aspects of the term IT governance.

Weill, Ross [21] focus the responsibilities and the decisions rights and they follow a static and structurally oriented viewpoint on IT governance. They define: “Specifying the decision rights and accountability framework to encourage desirable behaviour in the use of IT. IT governance is not about making specific IT decision – management does that – but rather determines who systematically makes and contributes to those decisions.”

The IT Governance Institute (ITGI) underlines on the one hand the responsibility of the top management but on the other hand it emphasizes the meaning of organizational structures and processes: „IT Governance is the responsibility of the Board of Directors and executive management. It is an integral part of enterprise governance and consists of the leadership and organisational structures and processes that ensure that the organisations’ IT sustains and extends the organisations’ strategy and objectives.”[9]
Grembergen et al. [8] define: “IT governance is the organizational capacity exercised by the Board, executive management and IT management to control the formulation and implementation of IT strategy and in this way ensure the fusion of business and IT.” They focus amongst others the Business/IT-Alignment, and more than ITGI, refer to a dynamic viewpoint of IT governance and the methodological process oriented support.

The metamodel we present in section IV has to capture both: the more static and structural oriented as well as the dynamic methodological viewpoint.

3. Models and Metamodels

In IS research we use models to abstract from real world phenomena and real world objects. The representation as a model usually is the first step of developing an application or software system.

If the subjects of research are models, and not the real world or the universe of discourse, we create models of models. Usually a “model of a model”, which is a higher level abstraction, is called metamodel.

Going from the instance level (real world, UoD) to the model level and further to the metamodel level denotes the application of abstraction mechanisms. The most used mechanism in computer science and information systems seems to be the linguistic abstraction.

In order to describe the abstract syntax of a language, usually linguistic metamodels are applied. They define the available language elements (modelling constructs) and relationships between them (e.g. constraints), as well as their meaning and generative rules [7].

The abstract syntax of a modelling language defines rules for structuring the real world (or some part of it) which it should describe by specifying the building blocks (phenomena) the world consists of, according to the respective language. For example, Chens E/R-model defines that the world (the universe of discourse) consists of entities, relationships and attributes. The representation of the subject is matter to the concrete syntax which defines the assignment of abstract syntax elements and their relationships to representational objects (e.g. symbols). The representation, e.g. the shape of the symbols used, is called notation. A modelling language usually has one abstract syntax but may have several notations (e.g. see the manifold of E/R-model notations).

It is necessary to stress that linguistic metamodeling is not the only way to perform metamodeling because various mechanisms of abstraction can be used for different purpose and applications. Karagiannis and Höffner [12, 13] e.g. emphasis the ‘ontological metamodeling’. Other researchers also work on ‘linguistic or physical metamodeling’. In contrast to linguistic metamodeling, the ontological metamodels deal with the classification of model elements according to their content [1], [2].

In order to metamodel governance frameworks we will describe relevant model components. We do this by abstracting from instances using the classification abstraction (‘instance-of’) and we extract the relevant content.

As metamodels represent the “deep structures” of a model, the language used to formulate the model (the metalanguage) must be able to represent the concepts of the model. Here, a dialect of the E/R-approach (see Chen, 1976), the extended E/R-model (eERM), is used [18, 19]. Therefore, we focus on the static aspects of the framework and are not able to capture e.g. information flows directly.

4. Best Practice Frameworks

4.1 Basics

As mentioned in the introduction, science offers little guidance to IT management and IT governance issues. Therefore, in the last ten years a range of open best practice frameworks (ITIL, COBIT) as well as proprietary frameworks were developed (Microsoft Operations Framework (MOF), IT-Service-Management (ITSM) of Hewlett-Packard, or the IBM IT Process Model (ITPM)).

These best practice frameworks which are also subsumed under the developing topic “IT governance” describe goals, processes and organizational aspects of IT management and control. They are created in practice and are given to use in practice. One point regarding the development of best practice models is very interesting: practitioners from the business world consolidate their knowledge aiming to define generally accepted rules, processes, and characteristics. Despite the fact that scientists also participate in the development of already mentioned frameworks such as COBIT, especially practitioners are members of the relevant committees and boards.
From an academic point of view, these best practice frameworks can be seen as an interesting object of research, not only because the models are widely spread in practice but also because they incorporate a huge amount of consolidated knowledge. As mentioned before, a sound scientific discussion and foundation of these models is missing but could be fruitful.

4.2 COBIT
In the following we solely focus on COBIT (Control Objectives for Information and Related Technology). COBIT describes a generic process model, that defines relevant processes and activities which one should find – according to the idea of best practice – in an IT department or organization. Whereas earlier versions put the main focus on IT audits, the COBIT framework meanwhile developed to a full-blown support of IT management covering most relevant tasks and areas of this topic. In a macro-perspective the IT processes are arranged by grouping them into four so called control areas, which are structured similar to the well known Deming/life cycle (Plan, Do, Check, Act) (figure 1).

For each of the 34 IT processes various components, such as business requirements and IT goals are defined, which satisfy them, controls and metrics as well as activities resources, responsibilities, etc. (figure 2).

Furthermore, there are other components and specification for one process, e.g. the persons who should be informed, should be consulted etc. are defined. These parts will be introduced in 4.3 with the metamodel.

4.3 The Metamodel of COBIT
Mainly, there were two reasons to start with metamodeling COBIT. First of all, this framework is well structured in chapters and components, and therefore closed in itself and self-contained. Secondly, COBIT is holistic and represents (nearly) all tasks and processes an IT organization should carry out.

For example, ITIL [16] is – like COBIT – holistic, but has a lack of structure. On the other hand, e.g. CMMI [20] focuses on a specific task (development), but has a coherent structure.

However, these existing structures primarily serve the purpose to present the framework consistently and structured. It supports the navigation and the usage of the framework but may not be mixed up with a metamodel.
A goal of metamodelling the framework is to extract and present the underlying logical and semantically rich relationships.

Here we use an abstraction mechanism which extracts the components of the underlying model. We use the well known extended E/R notation to represent our version of the COBIT metamodel. The analysis is stepwise and takes place in fragments which are in the end combined to one model. Initial point of the partial analysis is the entity type ‘process’ and thus, it is also the later necessary entity which integrates the fragments.

4.3.1 Control Objectives, Activities and Results

In COBIT, 34 IT processes are presented which produce one or more outputs which vice versa are used as inputs in other processes. Input and output are results. According to this, the entity type result ‘is-a’ output or input of a process. Typical results on instance level are documents like reports on costs, risks or plans on IT-strategy.

Moreover, a process consists of control objectives which are statements of desired results or purposes to be achieved by implementing control procedures in a particular process. These control procedures should provide ‘reasonable assurance’, that business objectives will be achieved.

Furthermore, a process includes activities, which give a detailed description of what is done. These activities are carried out by specific persons like the CFO, the CIO, or an architect. Therefore, we link activities to the concept role (fig. 2).

4.3.2 Goals and Metrics

Each process of the framework has goals, which can be divided into business goals, IT goals, process goals and activity goals. The goals again are in relationship with each other. Thus, IT goals activate process goals, which in turn end up in activity goals (e.g. IT goals define what the business expects from IT; Process goals define what the IT process must deliver to support IT’s objectives etc.). Each goal is measured with the aid of different metrics (key goal indicators and key performance indicators).

Furthermore, a process contains information criteria, which are abstract business goals. The information criteria proposed by COBIT are effectiveness, efficiency, confidentiality, availability, compliance and reliability. For every process COBIT states if these criteria are supported. It is distinguished between a primary and a secondary relationship.

Goals as well as metrics usually are neither considered as components in method descriptions [5] nor in the widespread modelling notations like EPC ((Event-driven Process Chain) or BPMN (Business Process Modelling Notation) [14]. From an IT governance point of view, goals and metrics are of high importance because in order to control, govern and manage, you have to quantify the relevant facts (“You cannot manage without measuring”) (fig 3).

4.3.3 Maturity Model, IT Resource, Domain

Each process is assigned to one of four domains, which are arranged according to the life cycle. Further components of COBIT are a maturity model, four domains and IT resources. Each process can be assessed by a maturity model to determine its level of maturation. This is the starting point for a continuous process improvement of the process maturity and its controls.

In order to achieve any results, a process needs the entity type IT resource. Implicit components as the life cycle orientation of COBIT could enter the metamodel as principles. However, a principle can
not be dedicated to a single entity type. Implicit basic principles form the framework as a whole and thus have to be put in another level of the metamodel.

4.3.4 IT Governance Focus Areas

Finally, each process has the attributes process code and process description. The process code is a unique identifier of the process. It consists of the abbreviation of the domain and a number.

Furthermore, each process supports a specific IT governance focus area. These IT governance focus areas 'describe the topics that executive management needs to address to govern IT within their enterprises’ [10]. For each process there is an indication if it addresses the focus area. Like above it is distinguished between a primary and a secondary relationship.

Figure 4 shows the integrated metamodel. The entity type process is used for the integration of the partial models presented above.

To conclude, by building the metamodel of COBIT, a lot of components could be extracted as being important for IT governance. A support for IT governance has to define certain activities (or processes) which generate specific results. The process aspect is more closely related to the dynamic and control oriented view on IT governance. From the static/structural viewpoint, it is of primary importance to organize work and to assign responsibilities that is to relate roles to processes or activities. For IT governance purpose, the assignment of roles to results might also be required sometimes. This aspect is missing until now in fig. 4. For both viewpoints, the assignment of metrics and goals to processes is of outstanding importance.

5. Application and Usage of the Metamodel

Several advantages accrue from representing IT governance frameworks like COBIT, ITIL or CMMI as method metamodels. In the following we will discuss some of the resulting benefits and possible applications.

First, the representation allows the comparison of different frameworks on an abstract level. Once the components are extracted, frameworks can be examined and analyzed. Thus, other frameworks can be checked for completeness with the aid of the metamodel. Accordingly, one can deduce that ITIL – in contrast to COBIT - does not provide metrics and other components for assessment to the extent COBIT does. Another benefit of the metamodel is the integration of new or existing processes in the COBIT framework.

![Fig 4 The COBIT Framework as a Metamodel](image-url)
This becomes apparent in the following example: The area outsourcing is hardly represented in the COBIT framework. However, outsourcing is an essential component of their IT strategy for some companies. With the aid of the metamodel a ‘Control of the outsourcing’-process can be developed under guidance. In order to develop this process, the metamodel has to be instantiated. In addition, the integration of the process into other existing IT processes can for example occur by linking the results. When inputs flow to the process and the output is used elsewhere, the new process becomes part of the overall IT process landscape.

One step further could be the metamodel based fusion of frameworks like COBIT, ITIL and CMMI. This might be of importance if one framework covers aspects, which are missing in another one. E.g. the new developments of the SEI (CMMI for Services; CMMI for Acquisition) can be a complement for COBIT. A metamodel based integration will allow a closer fit and can guide the models amalgamation on lower level.

Besides, the metamodel can be the starting point for the representation of COBIT in an application system. The components and the logical and semantic relationships are necessary, e.g. for the implementation in a semantic network. We are currently developing a framework representation with this technology which allows the flexible navigation within framework structures and the implementation of various views over the components (see Appendix).

This can be demonstrated by the screenshot: In the figure, PO1 (Define a strategic IT plan) is linked to PO 5 (Manage the IT investment) through various results (tactical IT plan, IT project portfolio, IT project portfolio). Furthermore, there are metrics which measure performance and the achievement of goals (key goal indicators, performance indicators). With a tool like this, we hope to support the implementation of governance frameworks in practice significantly.

6. Conclusions and Future Research

In this paper we discussed and presented a way to represent the popular IT governance framework COBIT as a metamodel. The intention was to demonstrate that metamodelling is a useful technique to gain a theoretical foundation. From our point of view, it is possible and fruitful to interpret IT governance frameworks as metamodels. IT governance models can learn from a rigid formalization and a systematic approach.

In the article, we extracted the relevant components performing some kind of ‘framework re-engineering’ on COBIT. The resulting metamodel brings some benefits for comparing and integrating different frameworks. Furthermore, frameworks can be checked for completeness against the model. An interesting application might be the representation of IT governance frameworks in a tool which was demonstrated in the previous section.

To give a widespread and holistic support for IT governance, it’s not enough to metamodel one framework. Instead, it is necessary to complement it with the knowledge of other frameworks and the findings of academic research. Therefore, the representation of COBIT can only be the first step in the process of building a wider metamodel.

Another interesting area of development is the situation specific and enterprise specific adaptation and configuration of governance models, because frameworks like COBIT and ITIL are seldom implemented completely and without modification. Metamodelling is the starting point for a methodological support for model adaptation. In further research, the metamodel presented should be made configurable by introducing and modelling variability on instance as well as on metamodel level.

References


Appendix: COBIT as a Semantic Net