Cost Oriented Modelling of IT-Landscapes: Generic Language Concepts of a Domain Specific Language

Lutz Kirchner

Institut für Informatik und Wirtschaftsinformatik
Lehrstuhl für Wirtschaftsinformatik und Unternehmensmodellierung
Universität Duisburg-Esssen, Campus Essen
Universitätsstr. 9
45141 Essen
lutz.kirchner@uni-duisburg-essen.de

Abstract: The accounting of IT-costs is an important aspect not only in the context of corporate cost accounting in general, but also for the management of IT. Various approaches to IT-cost accounting exist. However, they usually are implemented with only loose integration to the company´s IT-management. This often causes discrepancies between the numbers used for accounting and those for planning and operating the IT. A tighter integration of the information basis, which the two corporate functions (cost accounting and IT-management) share, is suggested, to solve this problem. One way to achieve this integration, is the implementation of cost accounting concepts in an IT-management method.

In this paper we introduce concepts for modelling IT-costs within a domain specific language for the modelling of IT-landscapes. The language is part of an IT-management method, which is currently under development. Using these concepts, costs can be modelled on the same level of abstraction as IT-components, thereby establishing a higher integration level between corporate cost accounting and IT-management.

1 Introduction

Information Technology (IT) in an enterprise generates a large amount of costs, which tend to rise steadily (cf. e.g. Horv02], p.741). To identify the sources of these costs as a prerequisite for recovering them from the users of the IT, as well as to provide a cost transparency, an IT-cost accounting has to be implemented. The services, which are provided by IT, are usually not only accessed by external, but also by internal customers, which potentially includes every division and department of a company. Due to the complex interrelations between the concerned IT-components, customers, services and business processes, it is difficult to determine, which approach to IT-cost accounting should be adopted. This decision is further influenced by the positioning of the IT-department in a company. The question has to be answered, whether a company sells IT-services to internal and external customers in order to make profit (*Profit Center*), just recovers the IT-costs from its customers (*Recovery Center*), or does not charge the costs at all, but only accounts for them (*Cost Center*). We cannot offer a solution to this problem inside the scope of this paper, since it strongly depends on the concrete enterprise. This holds for the determination of an appropriate level of detail for cost accounting, as well. However, aside from this decision, the actual applied approach should be compatible with the corporate IT-management, since cost accounting of IT is an important part of that task. Pragmatic approaches to IT-management, like ITIL[1], usually consider accounting and charging in their documentation on a rather abstract level. Scientific approaches largely omit cost aspects in their concepts (cf. section 3). Hence, it is motivated to design concepts, which aim at allowing for an accounting of IT-costs and are compatible with an IT-management method. This approach considers cost aspects within the context of tasks, like planning, evaluation, or optimisation of the corporate IT, on a common level of abstraction. Thus, IT-related decision processes as well as the valuation of an existing IT-landscape[2] can be supported.

In this paper we introduce concepts as part of a domain specific modelling language (DSL), dedicated to the modelling of IT-landscapes. The language is complemented with generic concepts, which can be used to model IT-cost aspects. This allows for the use of a model of an IT-landscape as input for common cost accounting methods. In the next section, we identify the concepts, which are required to model IT-cost aspects. Following, we evaluate selected approaches to the modelling of IT-landscapes with respect to possible concepts for the depiction of cost aspects. Thereafter, we introduce the *IT Modelling Language* (ITML), which is included in an IT-management method. The method is still work in process and is developed as a part of an enterprise modelling method, MEMO[3]. We present an excerpt from the meta-model of that language, which illustrates the concepts for modelling cost aspects. After a short application example and a discussion of the existing restrictions of the approach, we conclude with an outlook to future work.

2 IT Cost Accounting

Traditional costing methods (e.g. full and direct costing) usually differentiate between direct and indirect costs (cf. [Eis99], pp. 626, [Frie04], pp. 66). Regarding IT, if IT-components are exclusively used by one customer (e.g. one department), the according IT-costs can simply be attributed to this department (cost centre) as direct costs. If the IT is shared between several customers, the IT-costs are indirect costs and apportioned to the customers relative to the degree of utilisation. Cost units, like CPU-time, used disk space, number of printed pages, number of online accesses or employee hours (cf. [Horv02], p. 740), can be used to measure the IT usage. Additionally, the cost types have to be identified. Typical IT-related cost types are hardware, software, people, accommodation, external service and transfers (cf. e.g. [ITIL03]), which each can be

[1] *Information Technology Information Library*. For general information about ITIL see http://www.itil.co.uk, or e.g. [ViGu04] and [Olbr04].
[2] We use the notion *IT landscape* for the entirety of the corporate application landscape, the underlying IT infrastructure and their interrelations.
[3] *Multi Perspective Enterprise Modelling*, see section 4

divided in subordinated cost elements. External service costs are caused by the use of services provided by external parties. Transfers refer to the use of services or the acquisition of hard- and software from other departments. In [Bart02], p. 59, *people* is interpreted as *employee hours* invested in server operating, application support, repairs, upgrading and development of software. In [Kirs99], p. 136, software costs are further differentiated in costs regarding planning, design, acquisition, development, startup, operating and maintenance of software. ITIL proposes traditional cost accounting within its financial management module, offering a number of predefined cost types and cost elements, as well as a short overview of cost accounting. It is differentiated between capital costs (e.g. hard- and software acquisition costs) and operational costs (e.g. staff, maintenance). Beside cost types and cost units, cost centres, which are charged for their use of the IT, can be identified in traditional cost accounting. Typical cost centres are external customers or departments.

Horváth suggests an *activity based costing* approach (ABC, see [KaCo99], or *Prozesskostenrechnung*[4]) in the area of IT-cost accounting (cf. [Horv02], p. 741). This more recent costing method aims at identifying the relations between resources (e.g. IT-components), processes and products (e.g. IT-services) and emphasises the impact of process repetition on resource costs (cf. [Maen95], pp. 15). For this purpose, cost drivers are identified (cf. [Brau96], p. 53). A cost driver is a value, which quantifies the repetitive activities necessary to produce a process output (cf. [KuLo95], p. 90). An IT-cost driver can e.g. be the number of usages of an IT-service, consumed CPU-time or the amount of used disk storage, which can be used as a basis for charging the service.

A frequently used approach for determining the overall costs, the ownership of IT causes, is *Total Cost of Ownership* (TCO), proposed by the Gartner Group in the Eighties (see e.g. [Krcm00], pp. 181, [GRB04], pp. 499). In TCO, not only the initial acquisition costs of IT are accounted for, but also other, less obvious lifecycle costs (cf. [Krcm00], pp. 181). The latter are called non-budgeted costs, because they do not appear in the “book” in contrast to the budgeted costs. Some fundamental budgeted costs are related to hardware and software procurement, software development, communication, depreciation, support, administration, operation, installation, optimisation and maintenance of IT. Non-budgeted costs, which are very difficult to quantify, can be caused by negative productivity effects (delays, lack of ergonomics), system outages or a lack of user qualification, among others. TCO aims at making IT-costs more transparent to the user. It can be used as a planning tool by applying it to evaluate projected investments in IT, as well.

There exist a number of further approaches to IT-cost accounting, whereas the utilised cost elements are largely the same. Hence, in the following the insights gained by reviewing the selected accounting approaches are being used to formulate requirements for concepts for modelling IT-costs. To implement a basic support of IT-accounting in a modelling language, the following aspects have to be considered:

---

[4] The German *Prozesskostenrechnung* is also activity based, but differs in some aspects from the ABC. However, these differences have no impact on the topic of this paper and therefore are neglected in the following.

- IT-Assets
- Personnel
- Cost Centres
- Cost Types
- Cost Units
- Additional ABC Aspects
  - Business Processes
  - Services
  - Cost Drivers

IT-Assets are basically hard- and software, which are used to provide the services and which generate costs. The employees (personnel) perform maintenance and support, or use the IT respectively. Internal and external cost centres are e.g. departments and customers in the form of people or other companies. Cost types and cost units have to be related to the assets in order to track costs. In consideration of ABC, business processes, services and cost drivers have to be taken into account, as well. These fundamental concepts should be implemented in a modelling language to allow for the modelling of cost aspects against the background of IT-cost accounting.

3 State of the Art

In this section we give a short overview of selected approaches to the modelling of IT-landscapes. We cover ARCUS, $3LGM^2$, PMCSS and CIM, which all include a domain specific languages for this purpose. The evaluation of the approaches mainly focuses on language concepts, which are dedicated to the depiction of cost aspects. Despite the fact that ITIL considers cost accounting of IT, it does not provide any means for modelling IT-landscapes. Therefore, it is omitted in this section.

ARCUS: ARCUS[5] has been developed as a method for managing software architecture landscapes in an enterprise (see [HJT+99], [HMT02])[6]. It emerged as the result of a cooperation between the *F.A.S.T. GmbH*[7] and the *Bayerische Landesbank*[8]. The method comprises a modelling language and a process model. A model of an IT-landscape is structured into four layers: the *Business Process Architecture Layer*, the *Problem Domain Architecture Layer*, the *Logical Application Architecture Layer*, and the *System Architecture Layer*. Neighbouring layers are connected with each other through interlayer relations of model elements. The consideration of business processes, applications and hardware components in the modelling language potentially allows for managing cost aspects, which can be used as input for the costing methods discussed in section 1. However, concrete concepts for that purpose are not documented.

---

[5] The meaning of the acronym is not given in the available publications.
[6] In [HJT99], p. 2, the term *architecture management* is used and defined as a central service provider in an enterprise, that ensures the compatibility of utilised IT-systems and software.
[7] www.fast.de
[8] www.bayernlb.de

3LGM²: The 3LGM² (Three Layer Graph Based Meta-Model) is an enhancement of the 3LGM (Three Level Graph-based Model, [BAW+97]). It defines a modelling language for the description of Hospital Information Systems (HIS). A 3LGM²-model aims at supporting information management tasks, especially the planning and evaluation of information systems, in a hospital. A model consists of three different layers (see [WBW03a], [WBW03b] and [BHW+04]). These layers - or perspectives on the information system - are the *procedure layer* (in [WHB+04] it is designated *domain layer*), the *logical tool layer* and the *physical tool layer*. Neighbouring layers of a model are interconnected through relations of model elements belonging to different layers. As with ARCUS, the available abstractions on IT-landscapes are situated on separated layers. However, a *business process* abstraction is unaccounted for. Instead, the procedure layer provides a functional view on a HIS. Additionally, there is no support of economic aspects on a model yet. However, such enhancements are planned for future work (see [BHW+04]).

PMCSS: A method for modelling IT-landscapes has been introduced by J. Kirsch in his habilitation ([Kirs99]). The approach aims at extending ARIS with a component for the planning of IT-landscapes. The included language is defined by a meta-model in eERM (extended Entity Relationship Model), which is added to the existing ARIS concepts. The language allows for describing functions, processes, data and the organisational structure. The level of detail regarding the attributes defined in the meta-model, which serve to store mainly technical information about the IT, is high. In contrast, economic aspects are solely focused on tracking costs regarding planning, design, acquisition, development, startup, operating and maintenance of hard- and software (cf. section 2). These cost types are implemented as properties of particular types (like e.g. hardware components) and can as such not be modified according to the users needs.

CIM: The *Common Information Model* (CIM) published by the *Distributed Management Task Force* (DMTF)[9] comprises a meta-model (meta-schema) and a conceptual reference model (schema) for IT-landscapes. The meta-model defines the basic concepts, which are used in the reference model for the vendor-independent description of the components of an IT-landscape. The main design goal of CIM is the provision of a schema, which can be mapped to LDAP-directories or databases. A CIM-compatible database can serve as a basis for IT-management activities, especially the administration. CIM is purely a concept for describing IT-components. The main application scenario of CIM is using it as a schema for databases or LDAP-directories. Due to the absence of a *business process* abstraction as well as the negligence of cost aspects, it does not contribute to cost accounting.

Summary

The briefly introduced approaches all provide concepts for the modelling of IT-landscapes with a varying degree of detail. However, none of them provides adequate concepts for the depiction of cost aspects, which are needed to serve as a basis for an IT-cost accounting. Against the background of the shortcomings of these modelling

---

[9] The specification if the most recent version 2.9.1 can be downloaded from [CIM05].

approaches, we present a language for the modelling of IT-landscape, which provides basic concepts for modelling cost aspects.

4 The IT Modelling Language

In this section we introduce the domain specific IT Modelling Language (ITML). Beforehand, we want to briefly discuss some advantages, that a DSL holds compared to a general purpose language (GPL) in certain scenarios (cf. e.g. [EsJa01] and [LKT04]), in order to justify its utilisation in our context of modelling IT-landscapes. First, a GPL usually does not provide all required concepts for the creation of a model of an IT-landscape on an appropriate level of abstraction. Concepts, like *hardware device* or *application*, have to be invented by the modeller using the general purpose concepts (e.g. *class* in UML) of the given language. This implies that all the domain specific properties, like constraints and valid relations to other model elements, have to be specified by the modeller. Obviously, this proceeding comprises significant potential for conceptual errors or inconsistencies. Furthermore, the abstract depiction of the domain using a GPL potentially results in models, which are difficult to understand by anyone but the modeller. The concepts used for depiction most likely do not correspond to the *Weltanschauung* of possible addresses, and therefore may not be as helpful to serve as a basis for a discussion about the depicted domain, as required[10]. Another advantage of an elaborated meta-model of a DSL as the core language definition, is that modelling tools can check the validity and integrity of any model formulated in the given dedicated language on the basis of that meta-model[11].

The ITML is part of a modelling method, which aims at supporting tasks related to the management of IT, as they are part of any corporate *IT-management*, *information management* and *IT-controlling*[12]. These activities can be roughly categorised as *operational* and *strategic (*including *tactical).* Typical operational activities are the implementation of IT-landscapes and their administration. Strategic activities are the planning and design as well as the evaluation (including optimisation) of the corporate IT. An important prerequisite for a successful performing of the above mentioned activities is the provision of analysis concepts. Possible analysis tasks, which the method aims to cover, can be attributed to the categories data provision for inventory control and IT-cost allocation, analysis of the IT-architecture, detection of weak spots/bottlenecks, and detection of integration/redundancy issues. A model of an IT-landscape serves as a data repository, which holds the information about the IT, which is necessary to perform these analyses.

---

[10] Concepts that directly correspond to the viewer – or modeller - foster the clearness of a model (cf. [Fran97]).
[11] Cf. [FrPr97], p. 24.
[12] In the following selection of references, basic IT-management activities are discussed: [ZaBr03], pp.7-8, [StHa05]. p.427, [Horv02], p. 720, [GRB04], p. 392, [PMK04], pp. 82, [Stic01], p. 18, [Krcm00], p. 155 and [Hein99], p. 156.

The IT-management method is integrated in the enterprise modelling method *MEMO*[13]. By using the MEMO modelling concepts as a framework for the integration of the IT-landscape concepts, the available abstractions for the modelling of the organisational and operational structure of an enterprise can be reused. This approach promises a mutual enrichment of both methods. Chiefly due to the tight integration of the MEMO modelling languages[14], which provide specialised, intuitive concepts for the construction of multi-perspective enterprise models, MEMO is well suited for this task. Included languages are the *Strategy Modelling Language* (SML), the *Object Modelling Language* (OML), and the *Organisation Modelling Language* (OrgML). The latter language is of special interest in the context of this research paper. It chiefly serves to represent the organisational structure as well as the business processes of a company.

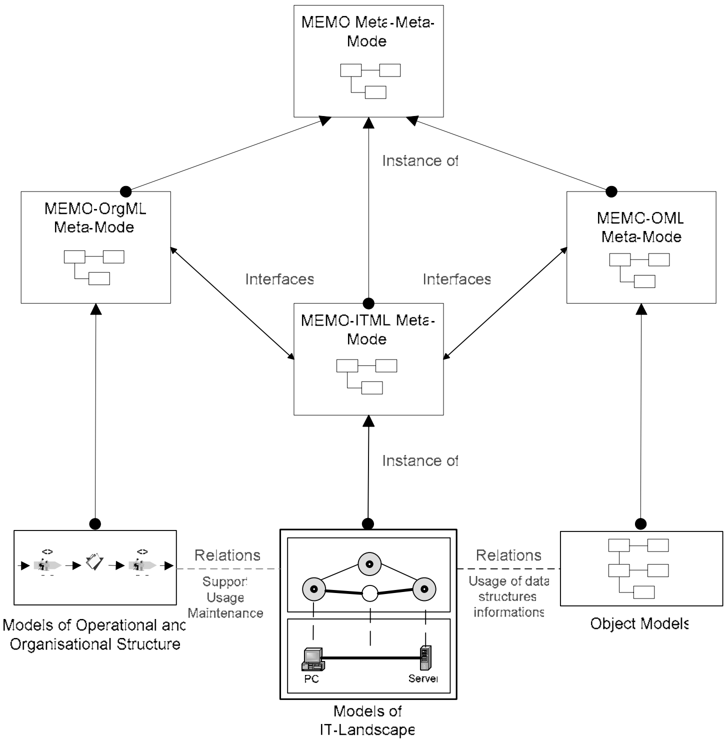

Figure 1: Integration of the ITML with the MEMO-Languages

4.1 Integration of the ITML into MEMO

Since the focus of the paper is on modelling concepts, we outline the integration of the modelling language with the available MEMO-languages, without further referring to

---

[13] For an overview of MEMO be referred to [Fran94] and [Fran02].
[14] More detailed descriptions of the MEMO languages can be found in [Fran98b], [Fran98c] and [Fran99].

the overall method. Figure 1 illustrates the interrelations and interfaces between the ITML[15] and the other MEMO languages. The MEMO-SML is omitted in the diagram, since the strategies are implemented by business processes and hence situated on a higher level of abstraction. The integration of the various modelling languages is accomplished through the specification of common concepts in the meta-models of the languages (cf. [Fran98a]). Hence, to integrate the ITML into MEMO, the meta-model of the ITML also has to provide concepts, which it has in common with the other languages. The definition of the meta-models is done with the MEMO meta-meta-model. This provides for a fundamental compatibility of the language concepts. The same concepts are being used to define the meta-model of the ITML. Instances of the ITML meta-model are the models, which are used to depict IT-landscapes. Through the common concepts in the meta-models, an IT-landscape model has interfaces to business process models (e.g. via the concept *business process*), organisational charts (e.g. via *OrganisationalRole*), and object models (via a *class*-like concept). This allows for the depiction of support relations between IT-components and business processes as well as the responsibilities of certain roles for hard- or software components. OML diagrams can be used to create object-oriented models of the information, which is being used or stored by applications. In the context of modelling cost aspects of IT-landscapes, relevant interface concepts are the *business process* and the *organisational role*.

4.2 Language Concepts for Modelling Cost Aspects

Figure 3 shows an excerpt from the meta-model of the ITML, which focuses on the language concepts required for modelling IT-costs. We use the UML-notation for the meta-model, whereas the semantics of the concepts correspond to those of the MEMO meta-meta-model (see [Fran98a]). All meta-types are direct or indirect subtypes of *Element* (see Figure 2) and therefore own the properties *name, description* and *numInstances*. The latter enables a type on model-level to track the number of its instances. Furthermore, a modeller can define his own value types or property types, which basically correspond to the tagged value and attribute concepts of the UML (cf. [OMG04]).

The meta-model in Figure 3 comprises the above mentioned interface concepts *business process* and *organisational role*. A business process is supported by services, which are provided by software applications. Applications are run on hardware devices, which together with the applications represent the IT-assets. In the complete meta-model the above meta-types are further elaborated to reflect the complexity of a corporate IT-landscape and the related business processes. However, these details are omitted in this paper, as are all other language concepts and properties, which do not directly contribute to the topic.

[15] In [Kirc03] an early design of the language is being introduced.

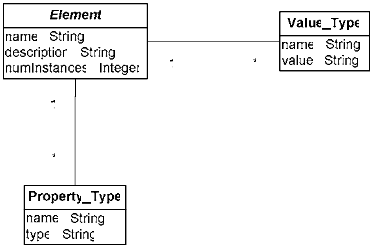

Figure 2: Core Elements of the ITML

The group of people, which are related to the business processes, e.g. as administrators or operational employees, can be modelled by using *PersonType*. Cost centres, e.g. departments or external customers, are implemented as *OrganisationalUnit* and *ExternalPartner*.

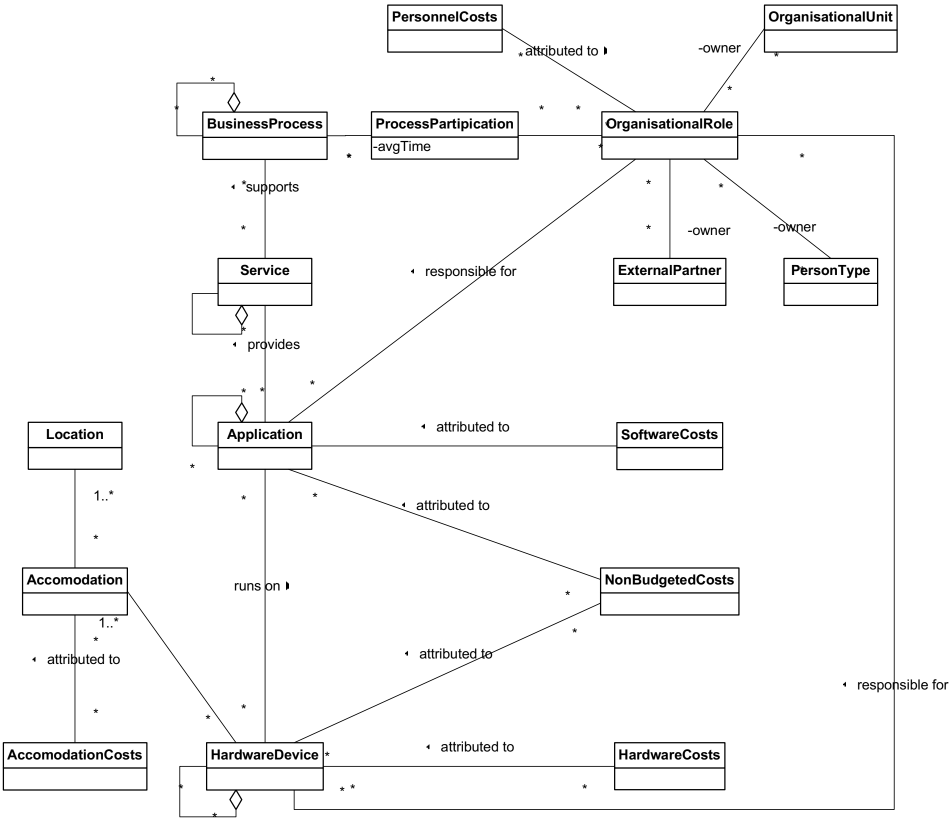

Figure 3: Excerpt from the ITML Meta-Model with Focus on Cost Concepts

The relation of organisational units, external partners and types of persons to a business process is realised by the meta-type *ProcessParticipation*, which is connected to the organisational role, the other types can play. Thus, properties of the relation can be encapsulated in a distinct relation type. To allow for a responsibility-relation between organisational units and applications or hardware devices respectively, without having to consider the concerned business processes, two associations *responsible for* are included. Additional concepts are *Location* and *Accommodation*, which represent the accommodation of hardware at a location.

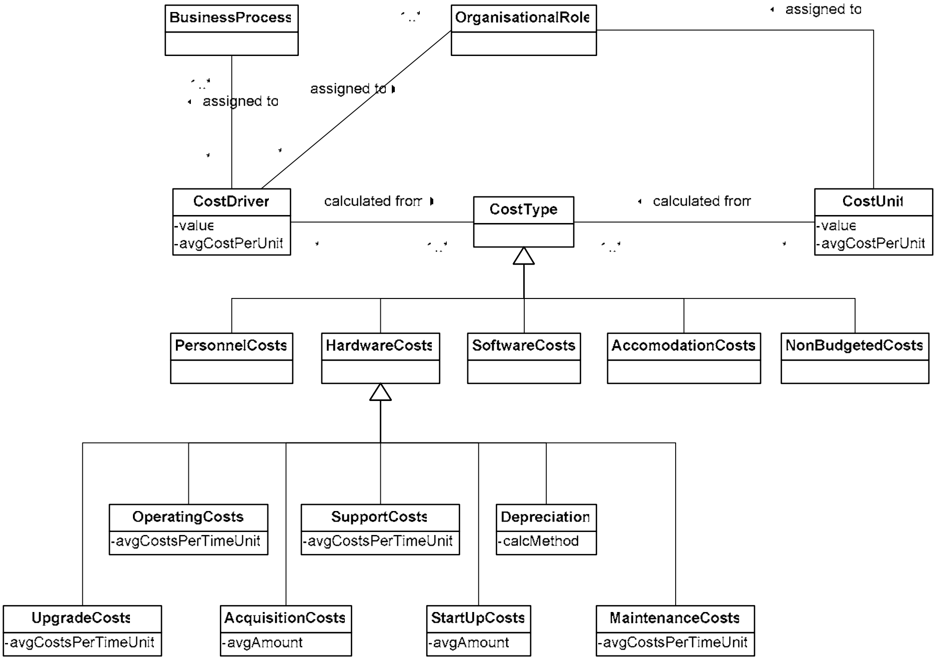

Figure 4: Interrelations of Cost Concepts

Also included in the meta-model are cost types, which are related to the respective elements, which generate the costs. Included cost types are personnel, hardware, software, accommodation and non-budgeted costs. Transfer and external service costs, which are mentioned in section 2, are not explicitly included, since they are aggregated costs of the other types. In Figure 4 the cost types are shown as specialisations of the abstract meta-type *CostType*. In general, the differentiation of the cost types on meta-level can be problematic, since the interpretation of their semantics is not standardised in the context of IT-cost accounting. Thus, the presented abstractions are subject to critical reviews and possible future modifications, if applicable.

Every cost type can be related to a number of cost units, which in turn are related to organisational roles. The owner of a role acts as a cost centre in this context. Considering ABC, a cost type can also be related to cost drivers. These are connected with organisational roles as well as business processes. The meta-type *HardwareCosts* is exemplarily specialised in a number of sub-types, which demonstrates a possible differentiation of that superordinated cost type. The specialisation of the other cost types is omitted in this figure. An orthogonal differentiation of the cost types (e.g. in capital and operational costs) on meta-level is not considered, since this can be implemented by the modeller on model-level according to the concrete corporate requirements.

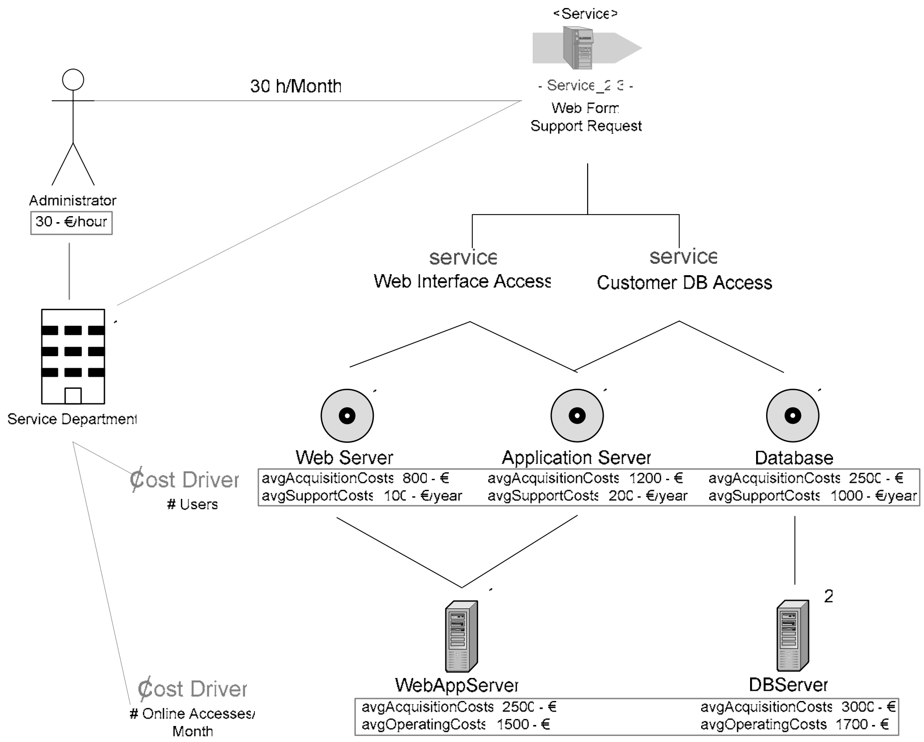

Figure 5: A Simple Example for Modelling Costs

Figure 5 shows an exemplary, simplified model, which demonstrates how costs can be complemented to the model of an IT-landscape. A business process *Web Form Support Request*, which is fully automated, raises IT-costs. The customers, which use the IT to access a web form, fill in a support request. The resulting data is stored in a database. The application types *web server*, *application server* and *database*, with one instance each, are attributed with specific average costs, belonging to the cost types *acquisition costs* and *support costs*. The applications provide the necessary services for the business process and run on hardware devices (a server for running web and application server software and two servers for the database). The hardware devices are also attributed with average costs (acquisition costs and operating costs). A number of administrators, who are involved for an average of 30 hours per month each, are attributed with an hourly wage rate and assigned to the process. The value of the cost driver instances, which are used to charge the cost centre, can be calculated on the basis of the IT-costs related to the execution of the business process and the concrete values (# Users and # Online Accesses/Month). In this example, the sole cost centre is the service department of the company.

The above introduced modelling concepts complement a language for the modelling of IT-landscapes with cost abstractions, which are required to perform an IT-cost accounting. Since the concepts are designed to be generic in a way, that they fit in various accounting approaches, the level of detail on meta-level is relatively low. Additionally, due to the fact that the resulting models, like the one in Figure 5, are situated on type-level, the given concrete values hold for all possible instances of a type. Thus, only average values can be expressed on this level of abstraction. An instance of the type *DBServer* e.g. can be expected to have average acquisition costs of 2500.- €, whereas the actual costs of the two possible instances may differ. With this restriction in mind, such a model may be used as a tool for planning and optimising IT-services with respect to the costs they generate. However, to effectively integrate the concepts with the corporate IT-accounting, concrete instances have to be included in the calculation. By creating reference models for specific cost accounting methods, which describe the necessary properties of concrete instances of an IT-landscape, an interface to accounting can be created. Furthermore, enterprise specific models can be created to cover more specific requirements. Thus, a tool which stores concrete data about the corporate IT on the basis of a reference model of the given language, can be used to extract the required accounting data.

5 Summary

In this paper we introduced concepts for modelling IT-costs, which are integrated in a language dedicated to modelling corporate IT-landscapes. We identified the basic concepts, which are required for the depiction of cost aspects, by reviewing common cost accounting methods. Following, selected approaches to the modelling of IT-landscapes were evaluated, whereas the focus was on possible included cost concepts. Since the approaches lack most of the required abstractions, it was motivated to include basic cost concepts in a modelling language, which is a part of an IT-management method.

The introduced concepts for modelling IT-costs allow for the attribution of costs of various cost types to IT-assets, like computers and software. Furthermore, cost drivers or cost units can be included in a model and assigned to business processes or cost centres. This allows for providing data required for planning and optimising IT-landscapes. However, concepts which describe properties of concrete instances of IT-components were not further discussed and will be introduced in a future paper. With these additional concepts, which e.g. can be provided in a reference model, the corporate IT-accounting can be further supported. Furthermore, we presently support only cost accounting methods, which do not require information about the benefit of IT or a business process. This excludes e.g. *Target Costing*, the *Breakeven Analysis*, or IT-controlling in general. The *benefit*-concept is not yet implemented, but is scheduled to be included in the course of future refinements of the language. Concepts for the depiction of risks, like they are used in the *Total Economic Impact* method, as well as of quality aspects have to be developed in the future, as well. This will allow for a broader range of application of the language and further improve the usability of the overall IT-management method. Also planned is the implementation of the language concepts with a meta-modelling tool.

6 Literature

[BAW+97] Buchauer, A.; Ammenwerth, E.; Winter, A.; Haux, R.: 3LGM: Method and Tool to support the Management of Heterogeneous Hospital Information Systems, In: Computers in Medicine, Polish society of Medical Informatics, Vol. 1**,** S. 77-82, Lodz, 1997

[Bart02] Barton, N.: Measuring I.T. Performance – From Functional Models to Service Models. In: Brown, A.; Remenyi, D. (Eds.): Proceedings of the Ninth European Conference on Information Technology Evaluation, S. 55-61, Paris, 2002

[BHW+04] Brigl, B.; Häber, A.; Wendt, T.; Winter, A.: Ein $3LGM^2$ Modell des Krankenhausinformationssystems des Universitätsklinikums Leipzig und seine Verwertbarkeit für das Informationsmanagement, In: Proceedings der Multikonferenz Wirtschaftsinformatik (MKWI 2004), Essen, 2004

[Brau96] Braun, S.: Die Prozeßkostenrechnung. 2. Aufl., Verlag Wissenschaft und Praxis, Sternenfels et al., 1996

[CIM05] Common Information Model – Preliminary Specification. DMF, http://www.dmtf.org/standards/cim/cim_schema_v291_prelim (letzter Zugriff 17.5.2005), 2005

[Eise99] Eisele, W.: Technik des betrieblichen Rechnungswesens. Vahlen, München, 1999

[EsJa01] Esser, R.; Jannek, J.W.: A Framework for Defining Domain-Specific Visual Languages, In: Proceedings of the Workshop on Domain Specific Visual Languages at OOPSLA 2001, Tampa Bay, 2001

[Frie04] Friedl, B.: Kostenrechnung. Oldenbourg, München et al., 2004

[Fran94] Frank, U.: Multiperspektivische Unternehmensmodellierung: Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. München, Oldenbourg, 1994

[Fran97] Frank, U.: Möglichkeiten und Grenzen einer objektorientierten Modellierungslehre. In: Tagungsband der STJIA'97, S. 96-102, Erfurt, 1997

[Fran98a] Frank, U.: The MEMO Meta-Metamodel**.** Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 9, Koblenz, 1998

[Fran98b] Frank, U.: The MEMO Object Modelling Language (MEMO-OML). Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 10, Koblenz, 1998

[Fran98c] Frank, U.: Applying the MEMO-OML: Guidelines and Examples**.** Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 11, Koblenz, 1998

[Fran99] Frank, U.: Memo: Visual Languages for Enterprise Modelling. Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 18, Koblenz, 1999

[Fran02] Frank, U.: Multi-Perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS-35), S. 10 ff., Honolulu, 2002

[FrPr97] Frank, U.; Prasse, M.: Ein Bezugsrahmen zur Beurteilung objektorientierter Modellierungssprachen - veranschaulicht am Beispiel vom OML und UML. Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 6, Koblenz, 1997

[GRB04] Grob, H.L.; Reepmeyer, J.-A.; Bensberg, F.: Einführung in die Wirtschaftsinformatik. Vahlen, München, 2004

[Hein99] Heinrich, L.J.: Informationsmanagement. 6. Aufl., Oldenbourg, München et al., 1999

[HJT+99] Hermanns, J.; Jänsch, C.; Tensi, T.; Toenniessen, F.: Architekturmanagement im Großunternehmen, In: OBJEKTspektrum 04, 1999

[HMT02] Heberling, M.; Maier, Ch., Tensi, T.: Visual Modeling and Managing the Software Architecture Landscape in a Large Enterprise by an Extension of the UML, In: Proceedings of the Second Workshop on Domain-Specific Visual Languages – OOPSLA 2002, Seattle, 2002

[Horv02] Horváth, P.: Controlling. 8. Aufl., Vahlen, München, 2002

[ITIL03] Service Delivery. CD-ROM Version, TSO, St. Crispins et al., 2003
[KaCo99] Kaplan, R.; Cooper, R.: Prozesskostenrechnung als Managementinstrument. Frankfurt/Main et al., campus, 1999
[Kirc03] Kirchner, L.: Eine Sprache für die Modellierung von IT-Landschaften: Anforderungen, Potenziale, zentrale Konzepte, In: Sinz, E.J.; Plaha, M.; Neckel, P. (Eds.): Proceedings zur Tagung Modellierung betrieblicher Informationssysteme - MobIS, S. 69-86, Bamberg, 2003
[Kirs99] Kirsch, J.: Prozessorientiertes Management von Client-Server-Systemen, Deutscher Universitätsverlag, 1999
[Krcm00] Krcmar, H.; Informationsmanagement. 2. Aufl., Springer, Berlin et al., 2000
[KuLo95] Küting, K.; Lorson, P.: Stand, Entwicklung und Grenzen der Prozeßkostenrechnung. In: Männel, W. (Eds.): Prozeßkostenrechnung. Gabler, Wiesbaden, 1995
[LKT04] Luoma, J.; Kelly, S.; Tolvanen, J.-P.: Defining Domain-Specific Modeling Languages: Collected Experiences. Proceedings of the 4th OOPSLA Workshop on Domain-Specific Modeling (DSM'04), Vancouver, British Columbia, Canada, 2004
[Maen95] Männel, W.: Zur Bedeutung der Prozesskostenrechnung. In: Männel, W. (Eds.): Prozeßkostenrechnung. Gabler, Wiesbaden, 1995
[Olbr04] Olbrich, A.: ITIL kompakt und verständlich. Vieweg, Wiesbaden, 2004
[OMG04] UML Superstructure Specification. Object Management Group, 2004
[PMK04] Pietsch, Th.; Martiny, L.; Klotz, M.: Strategisches Informationsmanagement. 4. Aufl., Erich Schmidt Verlag, Berlin et al., 2004
[StHa05] Stahlknecht, P.; Hasenkamp, U.: Einführung in die Wirtschaftsinformatik. 11. Aufl., Springer, Berlin et al., 2005
[Stic01] Stickel, E.: Informationsmanagement. Oldenbourg, München et al., 2001
[WBW03a] Winter, A.; Brigl, B.; Wendt, Th.: $3LGM^2$ Methode und Werkzeug zur Modellierung von Unternehmensarchitekturen im Krankenhaus, In: Proceedings der Frühjahrskonferenz 2003 des GI Arbeitskreises Enterprise Architecture, S. 20-32, St. Gallen, 2003
[WBW03b] Winter, A; Brigl, B; Wendt, T.: Modeling Hospital Information Systems (Part 1): The Revised Three-layer Graph-based Meta Model $3LGM^2$. Methods of Information in Medicine 42 (5), S. 544-551, 2003
[WHB+04] Wendt T, Häber A, Brigl B, Winter A (2004). Modeling Hospital Information Systems (Part 2): Using the 3LGM2 Tool for Modeling Patient Record Management. Methods of Information in Medicine 43(3), S. 256-67, 2004
[ViGu04] Victor, F.K.; Günther, H.: Optimiertes IT Management mit ITIL. Vieweg, Wiesbaden, 2004
[ZaBr03] Zarnekow, R.; Brenner, W.: Auf dem Weg zu einem produkt- und diensleistungsorientierten IT-Management. In: Brenner, W.; Meier, A.; Zarnekow, R. (Eds.): Strategisches IT-Management. Praxis der Wirtschaftsinformatik Nr. 232, HMD, S. 7-16, dpunkt, Heidelberg, 2003