Integration of Conceptual Models and Data Services Using Metamodeling

Hans-Georg Fill
University of Vienna
Vienna, Austria
E-Mail: hgf@dke.univie.ac.at

Dimitris Karagiannis
University of Vienna
Vienna, Austria
E-Mail: dk@dke.univie.ac.at

Christian Lichka
BOC AG
Vienna, Austria
Email: christian.lichka@boc-eu.com

***Abstract*—The continuous shift to concepts of service orientation is a fact - not only in the area of information and communication technologies but also in business environments. Model-based approaches have been identified as a suitable means to bring together business and technology. In this paper we describe an approach how knowledge about business requirements laid down in the form of metamodels can be aligned with concepts of service orientation in information technology. For that purpose we propose an evolutionary process to migrate from existing metamodels to semantically annotated services by applying metamodel slicing techniques. The resulting data services are then passed back to the business environment where they may be reused for implementation tasks or for the integration in business applications.**

I. Introduction

The upcoming of information and communication technologies that improve automation and connect people to the global labor markets has resulted in a shift of resources from manufacturing into knowledge-intensive service industries supporting manufacturing and innovation [1], [2]. These *service systems* as value-creation networks that are composed of people, technology, and organizations are complex structures that require the adaptation of traditional business models. A recent example is the adaptation of business models to the internet-of-things and increased product-level intelligence [3], [4]. Their formal representation and measurement is a grand challenge for the success of the services economy [5], [6].

To identify and represent the relevant factors for improvement and innovation, service systems are viewed from two generic perspectives. On the one hand a *bottom-up* perspective that takes into account the actual technical operation of a business and on the other hand a *top-down* perspective where the innovation takes place on a strategic management level that is then transformed into concrete technical services and products. According to this view, it can be found that the IT related research in the area of *services* is today largely focused on bottom-up approaches dealing with the technical issues of IT architectures. Several attempts have been and are made to describe services formally [7], realize them technically [8] or how they can interact and be made interoperable [9]. Comparatively few approaches focus on the alignment of business and services from a content-oriented, management perspective [10]. Previous approaches that take up this challenge, e.g. as proposed by IBM by using the Component Business Model or in the area of EAM, focus however primarily on the domain of IT management [11], [12].

A central question that arises in this regard concerns the *identification* of possible technical services for supporting businesses [5]. One approach to achieve this goal is by decoupling the business and IT related structures through the introduction of an integration layer between the business process architecture and the software architecture [13]. However, the focus on business processes may not be suitable for all types of enterprise applications that can make use of services. This includes for example services on the strategic and performance management layer as well as on the layer of IT infrastructure management that are not directly related to business processes. Furthermore, the introduction of additional constructs to express services on the business layer requires additional effort from the side of business users. Service constructs need to be understood and services for the support of the business processes need to be separately specified. It thus seems a promising approach to develop concepts for decomposing a particular domain and use the existing systems and concepts for the transition to service orientation [14].

In the domain of enterprise modeling, several modeling methods have been developed in the past that permit to structure and process business domain knowledge in a formal way [15]. Besides their traditional role of supporting human understanding, these methods serve as a basis for the interaction and configuration of IT systems [16]. We have thus come up with the idea to revert to formal enterprise modeling methods as a basis for integrating models and services. With the following contribution the following two research questions shall be answered: A. How can potential technical services be identified in metamodels of enterprise modeling methods? B. How can these services be aligned with enterprise model instances? Thereby it shall be focused on an evolutionary transition from existing business knowledge encoded in enterprise modeling methods to technology-oriented services.

The remainder of the paper is structured as follows: Section II will shortly present the fundamental assumptions from the area of meta modeling and services that will be used throughout the paper, section III will present the key contribution of the paper in the form of two types of metamodel-based service evolution followed by a usage scenario and an evaluation in sections IV and V. The paper ends with a discussion of related work and a conclusion.

II. Metamodeling and Services

The use of models is today an established technique to represent knowledge in both business and IT domains [8].

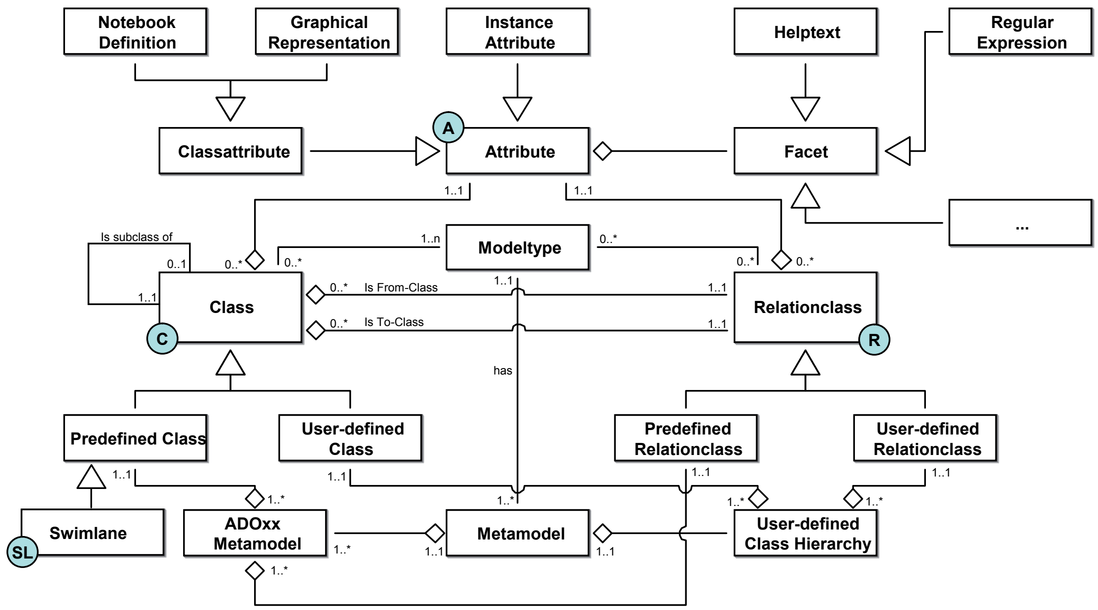

Fig. 1. ADOxx Meta-Metamodel (extended from [17])

Models represent a particular view on the world for the purpose of a user or a group of users. They are characterized by mapping, reduction, and pragmatic characteristics [18]. Models are created by using modeling methods, which consist of a modeling technique and mechanisms and algorithms [19]. The act of designing, formalizing, and developing a modeling method we denote here as *metamodeling*.

A. Metamodels

Metamodels are thereby used for specifying the modeling language [20]. More specifically, within the scope of this paper we regard metamodels as abstract conceptualizations of a specific user domain for the purpose of creating concrete models in this domain [15], [21]. For the IT-based processing of models it is required to formally describe these metamodels, i.e. the specifications of the syntax, semantics, and their visualization, i.e. the graphical notation [19], [22]. In the following we will primarily focus on the syntactic and semantic aspects.

To realize modeling languages it can be regarded as state-of-the-art to use meta-metamodels [19], [23]. Thereby the basic elements and relationships that can be used to define a modeling language are formally expressed. For the meta-metamodel it can either be referred to standards such as the Meta Object Facility (MOF) or proprietary concepts [17], [23].

The approach described here refers to a slightly extended version of the ADOxx Meta-Metamodel [17][1] (see figure 1): It provides as basic elements *classes* (C) and *relationclasses* (R) that may be further detailed by the addition of attributes (A). Furthermore, it contains concepts for *model types* for the definition of views on the class and relationclass instances together with attributes for the specification of the graphical representation. The meta-metamodel shown in figure 1 has been extended by a specific pre-defined class for the representation of swimlanes (SL) and visual markers that will be used in the following to visually express instantiation relationships from particular metamodels to elements of the meta-metamodel.

B. Services

Although information and communication technologies are today seen as the ”critical enabler” [1, p.39], the *science of services* takes a view that is broader than just the technical aspects [1], [5]. The crucial notion is that there exist both *business oriented* views as well as *technically oriented* views on services: Business oriented views usually do not take into account the technical interaction of different human or technology-based services but use high-level abstractions of the services’ functionality (e.g. [24]).

The discussion of technical services has taken place in the area of computer science for quite some time. Services in this context are regarded as loosely coupled components that encapsulate particular functionalities, are technology neutral and support location transparency [25]. The goal is to offer particular IT based functionality that can be easily composed to larger applications, re-used for different applications and invoked in a standardized way.

Technically oriented views consider two occurences of services: *Simple* and *composite* services. Simple services accomplish a specific business task whereas composite services involve the composition of existing services to access and combine information [25]. The goal of technical service oriented architectures (SOA) is to provide a framework for matching the needs of service consumers and the capabilities of service providers [26, p.8].

From the technical viewpoint the interface of a service is syntactically described in a standard referencable format in order to uniformely access its capabilities and interpret its responses [26, p.22]. This aspect is essential for the implementation and can be accomplished using open standards. To

[1] See also http://www.adoxx.org and http://www.omilab.org

enable legacy applications for service oriented computing they either have to be re-implemented or extended by SOA adaptors through wrapping techniques [27].

In both cases the functionality that is to be encapsulated by services as well as their interfaces have to be determined. In the next section it will be addressed how business oriented metamodels can be used to derive technical services and their interfaces thereby taking a step towards the business / IT alignment.

III. Identification of Data Services in Metamodels

Both in science and practice a number of metamodels for enterprise modeling methods have been designed in the past for supporting knowledge structuring and analysis. Examples can be found on all layers of an enterprise ranging from strategic and performance management to business process management, IT Management and many more [15]. Some of these approaches focus only on particular aspects of a business whereas others take holistic views of an enterprise, e.g. [28].

Traditionally, these metamodels do not incorporate particular service oriented concepts that would allow for a direct technical derivation of services. Although some of them contain concepts for services (e.g. [12], [29]) they often do not consider the technical interfaces in a way that permits to specify technical services. To transition to data services it can be chosen from two basic directions: a *redesign* that includes modifications and extensions of the metamodel concepts or an *evolutionary* approach that does not change the concepts of the metamodel but only modifies its structure.

The first direction is taken by a number of approaches that focus on the integration of data service concepts into the strategic, business process and application/IT layers [10]. Typically, such service concepts are *tightly coupled* with the metamodel and may be used as model elements on the model instance level. The drawback of this approach is however that it is not explicitly stated how data from actual services - e.g. a service delivering performance data - is integrated in the modeling environment, i.e. via the metamodel. Thus, it would not be clear for a developer of according data services feeding information into the modeling environment, how the data services are linked to the models.

To realize an approach that overcomes this limitation of a tightly coupled metamodel, we position metamodels in a different way. As shown in figure 2, we regard three layers: i. the *client layer*, which stands for all clients who wish to access some data sources. These can for example be humans, mobile devices, intelligent sensors as in the internet-of-things or other entities featuring some intelligent behavior. ii. the *technical layer*, which stands for the actual sources of data.

These can be of three types: a. *Structured data sources* such as enterprise resource planning systems (ERP), relational databases and the like. b. *Semi-structured data sources* such as schema-less data storages, e.g. key-value stores as used in many web applications today - see e.g. Facebook Graph API, Amazon API etc. c. *Unstructured data sources* which do not provide any structure and thus need to be processed specifically, e.g. by using natural language processing techniques. Typical examples for this latter type would be user posts in Facebook, product reviews on Amazon or textual documents in a virtual library. In between layer i. and ii. we position a *conceptual layer* in the form of domain metamodels. This layer interacts on the one side – as defined by a *data service access* – with the client layer. On the other side it interacts with the technical layer through a *data sources access*.

In addition to the mediation between clients and data sources, it can be decided in the conceptual layer how the usage of the data is to be envisaged. We thereby envison three scenarios. Either, the data retrieved from the data sources is necessary also for future use. Then it needs to be stored permanently. An example would be a recommender service that accesses social network data of a certain user group to conduct recommendations [30].

In a second scenario, the usage of the data is only necessary during the operation period of a service. In this case the retrieved data is only stored temporarily. An example would be a risk simulation service that requires information about the current impact and probability of risks for performing certain calculations [31], [32]. Upon completion of the calculations the data can be discarded. In a third scenario, the data is discarded immediately after it has been retrieved and handed to the client. An example would be an obfuscation service that transforms real names in models into abstract identifiers - after the identifier has been retrieved all original information needs to be discarded for privacy reasons [33].

In the following we present two forms for structure modifications of metamodels which we have developed for enabling metamodels to act as a conceptual layer between clients and data sources. We denote these modifications as *Metamodel-based Loose Coupling* and *Metamodel-based Semantic Coupling*. Both of them lead to the identification of particular domain services that can be integrated into model instances.

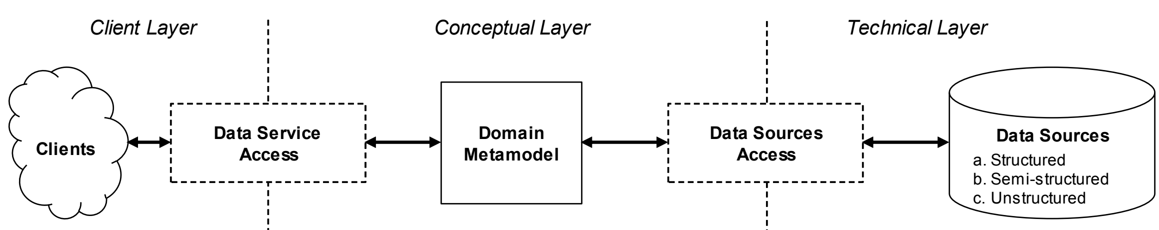

Fig. 2. Metamodels for Mediating between Data Service and Data Source Access Layers

A. Metamodel-based Loose Coupling

The main idea of Metamodel-based Loose Coupling lies in the decomposition of a particular metamodel into sub-metamodels and services - see figure 3. Thereby certain parts of the metamodel are defined as data services and separated from the remaining part of the metamodel on the level of relations - we denote this as *metamodel slicing*.

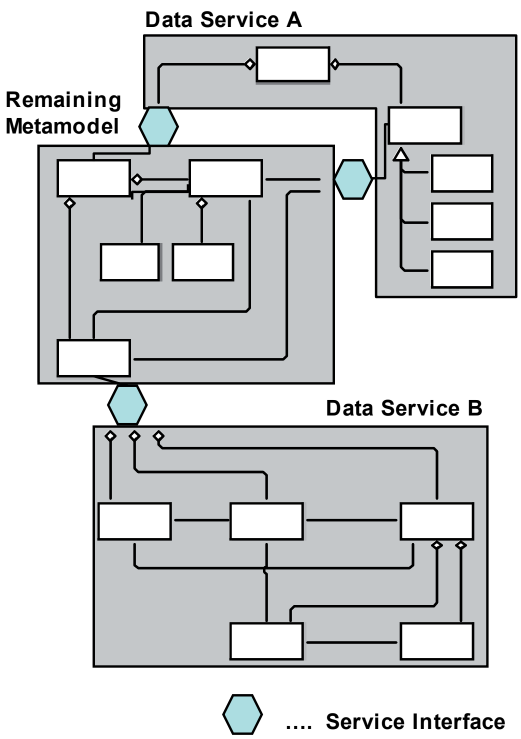

Fig. 3. Meta-Model Based Loose Coupling

The classes and relationclasses of the sub-metamodels are at first not changed. As there are however linkages between the split parts these have to be further analysed and incorporated as service interfaces. Through the derivation of the metamodels from a meta-metamodel, the linkages between the elements are sufficiently formally specified to allow for a direct mapping of links between the metamodel elements to these service interfaces.

More precisely, a metamodel class that conforms to a sub-class of a relationclass in the meta-metamodel contains a number of instance attributes as well as it participates in certain *Is From-Class* and *Is To-Class* relationships - cf. the meta-metamodel shown in figure 1. Thereby the cardinality restrictions between the relationclasses and the classes are specified. This information can also be used for defining the service interface: The part that is defined to be a service contains a subset of the original metamodel. The elements that are now part of the service subset S are then regarded as follows: Relation class instances within the service subset are left out and from the remaining classes in S only those are considered that participate either in an *Is From-Class* or *Is To-Class* relationship. As these are the classes that are directly required for the rest metamodel they constitute the service interface I. This means that the attributes of these classes become the properties of the objects exchanged with the derived service.

A relaxed formalization of these relationships can be represented as follows[2]: Consider a simplified version of a meta-metamodel $MM^2 = \{MT, C, RC\}$ that consists of sets of modeltypes $MT = \{M_1, M_2, \ldots, M_n\}$, classes $C = \{c_1, c_2, \ldots, c_o\}$, and relationclasses $RC = \{r_1, r_2, \ldots, r_p\}$. The modeltypes M_n comprise a subset of the classes and relationclasses, i.e. $M_n \subseteq (C \cup RC)$. For every relationclass r_p a pair of classes can be assigned to express the Is From-Class and Is To-Class relationships, i.e. $r_p \longrightarrow (c_a, c_b)$. For every class and every relationclass a set of attributes A is assigned, i.e. $\forall c_o \longrightarrow A^c_o$ and $\forall r_p \longrightarrow A^r_p$. A metamodel MM that is derived from MM^2 contains sets of model type, class and relationclass instances, $MM = \{MT^i, C^i, RC^i\}$. When the metamodel is sliced, a service $S \subset MM$ is created and the service interface I can be described as $I \subseteq S \backslash RC^i \in S \,|(\exists r_p \longrightarrow (c_a, c_b) \wedge c_a \in MM \backslash S \wedge c_b \in S)$ $\vee (\exists r_p \longrightarrow (c_a, c_b) \wedge c_b \in MM \backslash S \wedge c_a \in S)$. Based on the assumption that every class contained in I has assigned a set of attributes A, the interface I can be directly used for the concrete realization of IT implementations. This could be done e.g. by defining generic `get` and `set` operations for these attributes as properties.

B. Metamodel-based Semantic Coupling

While Metamodel-based Loose Coupling provides concepts to separate metamodel parts and derive certain services, it is not sufficient to achieve full independence of the original metamodel. Depending on the domain the metamodel has been designed for, a certain amount of interpretation may be necessary to assemble the services and the remaining metamodel parts. To go beyond the syntax-level interoperability that is established by the definition of services, it is a common approach to add formal knowledge representations [35]. Typically this is done in the form of ontologies which leads to three primary advantages according to Sheth et al. [36]: The *re-use and interoperability* among independently created and managed services is promoted; explicit formal and ontology-supported representations lead to *more automation* and the explicit modeling of the participating entities and their relationships allows to perform *deep and insightful analysis* [37]. Several standards are being actively developed that aim for the addition of semantics to web services [38], [39].

To enhance the described service identification based on metamodels by adding semantics, the following approach has been chosen [40]: Starting from the interface definitions I of the services, semantic annotations can be assigned via a semantic schema, e.g. an ontology (see figure 4). In a general sense, ontologies can be described as a set of classes C, object properties OP to express relationships between classes, datatype properties DP to assign data to classes and values V, i.e. $O = \{C, OP, DP, V\}$[3]. In contrast to metamodels ontologies usually contain large numbers of concepts that have been identified as relevant for a domain. The metamodel classes contained in I can now be mapped to classes of the ontology

[2] More elaborate formalizations could be given by using for example specialised formalisms such as FDMM [34].

[3] This corresponds to a highly simplified version of the Web Ontology Language (OWL)

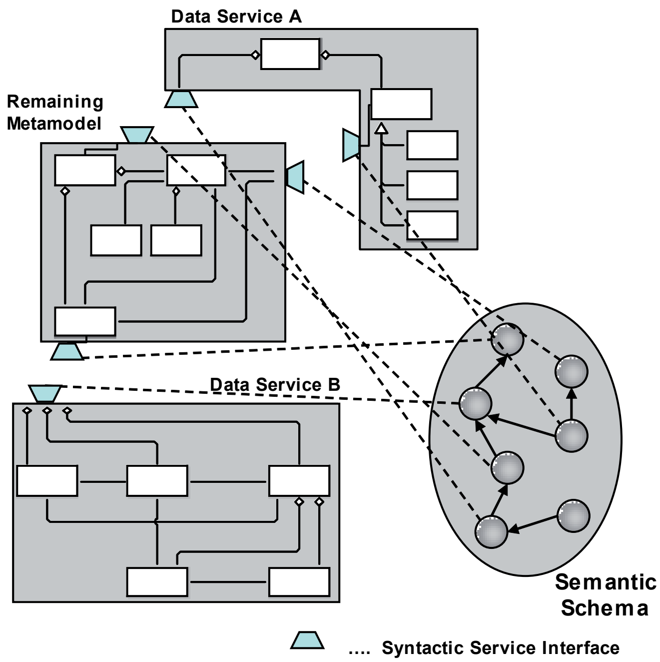

Fig. 4. Metamodel-based Semantic Coupling

and the relationclasses to object properties. In the same way the attributes in the metamodel classes can be mapped to datatypes in the ontology: $C^{metamodel} \longrightarrow C^{ontology}$, $RC \longrightarrow OP$, $A \longrightarrow DP$. This leads to a semantic annotation of the service interfaces. More detailed elaborations on how annotations with concepts from ontologies can be added to conceptual models are found in [35]. Through the use of the ontology the identified services can be independently processed and assembled. Furthermore, inferencing mechanisms working on the ontology can be applied to ensure consistency.

IV. Evaluation of a Concrete Usage Scenario

To illustrate the above described concepts and give a first evaluation we will revert to the domain of performance management. In this area, a high amount of flexibility is required in regard to innovation and adaptability [42]. Especially offensive strategies, that often comprise new management concepts implicate changes in management behaviour [43] and require immediate adaptation of relevant processes and IT together with performance measures [44], [45]. Thus, information must be reusable and adaptable in different contexts due to recently emerging requirements.

As a consequence, performance management concepts must be broken up into information service components with clear interface descriptions. The more it is possible to separate individual business services into service components, the easier the matching with IT service concepts can be done. Strategic concepts like SWOT analysis or BCG diagrams are considered to be business service concepts, reused in different business methods like Balanced Scorecard or Business Performance Management [28], [46]. The question now arises how appropriate IT-based management support systems can be designed on a conceptual level for supporting such scenarios [47].

A. *Running Example from Personal Performance Management*

To make these concepts easier graspable, we decided to use an example from a personal performance management project conducted with students a few years ago. In this project, students had to design their own balanced scorecards to assess their personal goals and achievements. For this purpose, they had to design cause-effect diagrams showing dependencies between their strategic goals, complement them with suitable performance indicators, and measure the achievement of the goals using concrete data. For example, the strategic goal to get a good job in the future was dependent on finishing their master program in business informatics and acquiring social skills. These social skills in turn were dependent on gaining experience in industry and engaging in industry networking. Possible performance indicators were for example the amount of their salary in their current job, the number of professional contacts in their social network, or the level of completion of their master program. The performance indicators could be directly calculated based on actual data, e.g. by retrieving payment data from their bank account statements or by counting how many business contacts they had collected in the past.

Although for such a small example also a pen and paper approach may be sufficient, the goal was to use state-of-the-art enterprise software to get insights into professional IT-based performance management. In this way the previously static balanced scorecards should be moved to an IT environment. By using IT services for different parts of the approach, model-based representations, e.g. for cause-effect diagrams and performance indicators, should be complemented with data services, e.g. for automating the retrieval of data from social networks or personal databases. The individual balanced scorecards should be easily adaptable to accomodate for emerging requirements, e.g. to take into account upcoming new job opportunities or changes in personal life preferences.

In the following we will illustrate based on this example how the above described approach for identifying services can be applied. This will be done from a conceptual level by using a metamodel. Subsequently it will be discussed how the actual technological realization is accomplished.

B. *Sample Metamodel for Performance Management*

As a first step it can be reverted to an existing metamodel from the area of strategic and performance management to formally characterize the domain [41]. Figure 5 shows an excerpt of this metamodel that contains the major classes, relationclasses, and the model types. Due to reasons of illustration, several additional dimensions have been left out in this excerpt (e.g. attribute assignments). The grey areas depict the assignment of elements to a modeltype. Thereby it is expressed that for example the *Indicator Model* type consists of the elements *Data Source*, *Calculation Algorithm* and *Performance Indicator* and several relations between them. The *Performance Indicator* element also participates in the cause-effect model and thus provides a linkage between the two model types. To further detail these elements, the references to the meta-metamodel have been inserted. Thereby it is defined that e.g. the *Data Source* and the *Calculation Algorithm* are expressed as an attribute (A) and the *Performance Indicator* as a class (C). The addition of the meta-metamodel references can

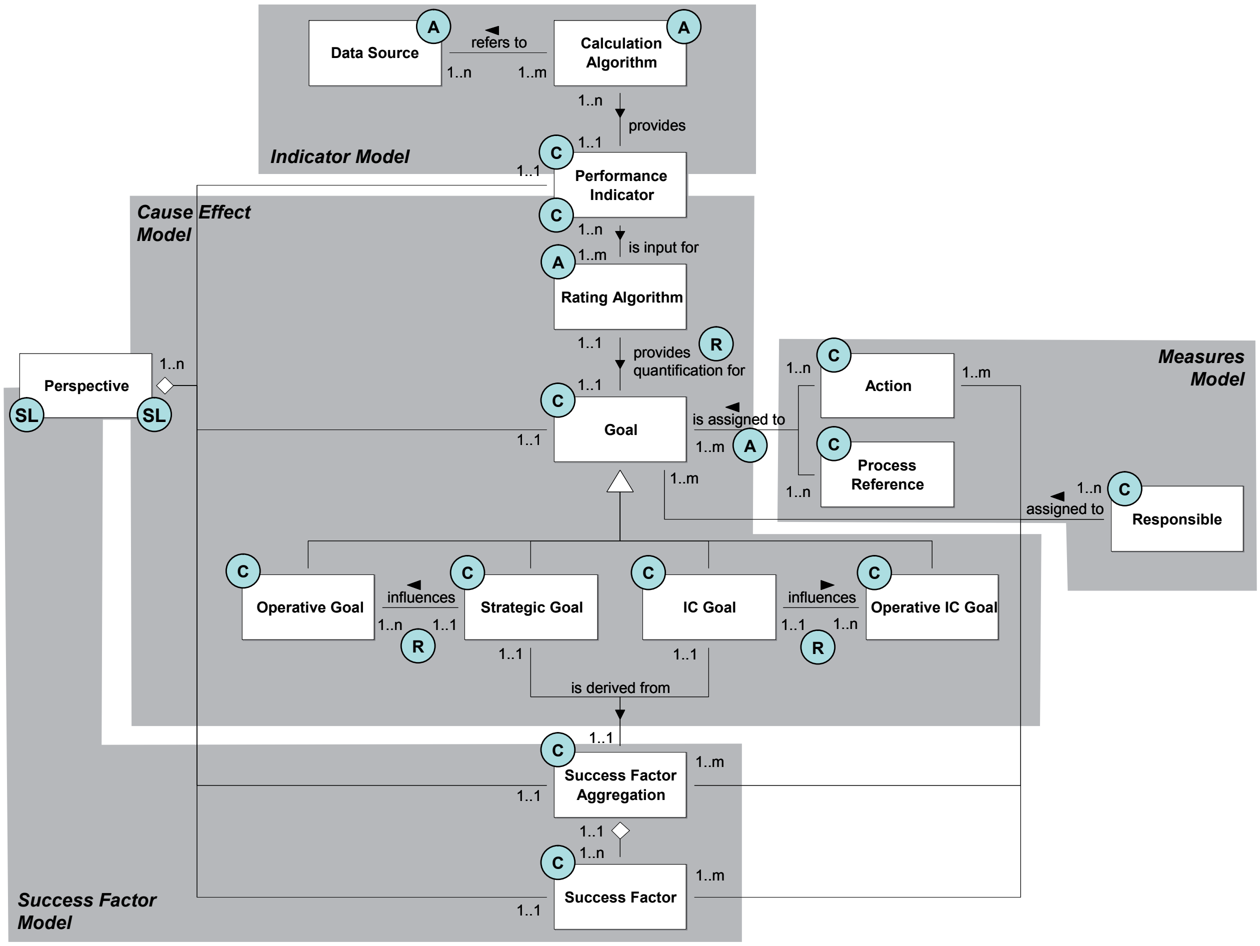

Fig. 5. Excerpt of a Metamodel in Strategic and Performance Management [41] with its Classes (C), Relationclasses (R), Attributes (A), and Swimlanes (SL)

also be regarded as a step towards a platform specific model (PSM) [48]. For a fully functional PSM and the implementation on a metamodeling platform more details such as the types of attributes would be necessary [49].

C. Transition to Data Services

With the availability of this metamodel it can now be determined which parts should be defined as data services. In figure 6 four possible data services have been identified. The slicing is independent of the definition of the model types although it may also be congruent with existing model types. This would allow for example to use the *Performance Indicator Service* shown in figure 6 as a direct input for the *Cause Effect Model* type shown above. According to the described derivation of service interfaces the *Performance Indicator Service* has then to deliver objects of the type *Performance Indicator* with the complete data structure in the form of assigned attributes. To be even more concrete, a performance indicator class may have attributes for the denomination of the indicator, the expression of its type (e.g. long-term or short-term oriented) or corresponding reference dates. With the availability of services for the different parts of the metamodel, the goal of re-usability as expressed for service oriented architectures is directly met. Upon availability of the implemented services, each of them can be immediately used for different application purposes without additional implementation efforts. However, the service interfaces are still being coupled to some degree to the original metamodel interfaces. The reason is that so far only the syntax of the interface has been made explicit but not the semantics.

The next evolution step is to map the service interfaces within the metamodels to an ontology and thus allow for a semantic mapping between the particular metamodel services. Based on an ontology, a direct linkage between the terms and concepts of the business context and the concrete IT services can be established. For the usage scenario at hand it could be reverted to various ontologies that have been proposed for the domain of strategic management and business administration - for an overview see e.g. [50]. The use of the ontology thus permits to handle semantic interoperability between different services. Thereby the function of the ontology as a semantic schema is to act as an intermediary between different interpretations for the services's interfaces.

In this way it could for example be expressed that the performance indicator concept of the performance indicator service and the performance indicator concept of the rating service are equal. In addition, also the attributes attached to the classes can be annotated individually, thus permitting even

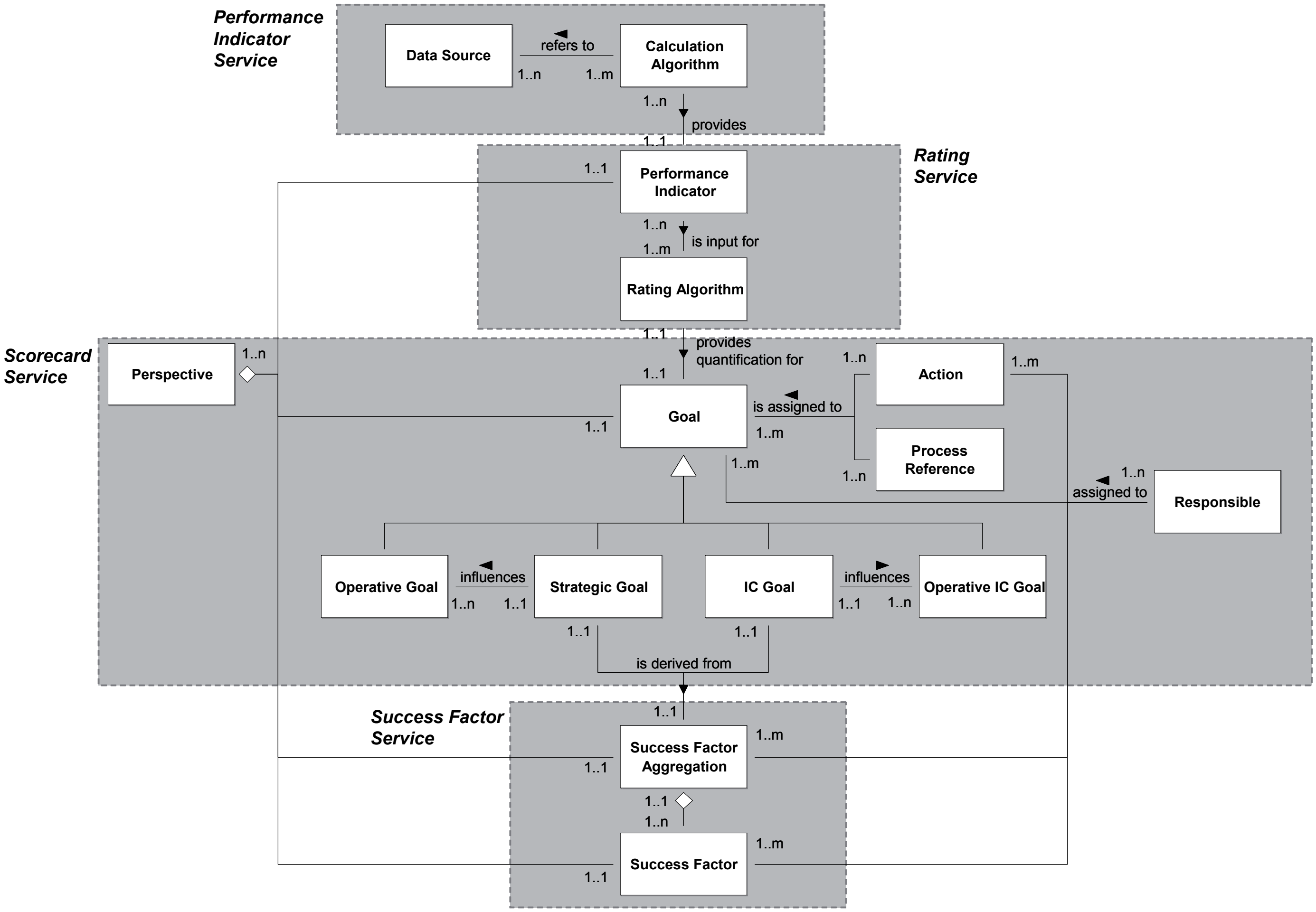

Fig. 6. Sliced Metamodel for Strategic and Performance Management

more detailed matchings. The equality thus not only refers to the syntactic level - i.e. that objects can be exchanged between the services - but that also the content transferred from one service to another matches semantically. This opens up a wide range of application scenarios.

First, it includes all technical scenarios that have been previously discussed for the area of semantic web services, e.g. [36], [38]. In addition, this also has an impact on the business level. The availability of such modular components that can be easily arranged on a conceptual level and which at the same time correspond to a detailed technical description, could also be used for easily specifying new business models. It thus becomes possible to adapt and evaluate business models that rely on explicitly defined IT services [51]. A direct application area can be found in the domain of the internet-of-things where several kinds of upcoming technologies such as intelligent sensors and products provide information that needs to be adequately integrated when designing new business models [3].

D. Possible Technical Realization

To illustrate the technical realization of the services based on the sliced metamodel we revert to the ADOscore performance management toolkit[4]. ADOscore offers an implementation of a metamodel for strategy and performance management that is similar to the one in [41]. Based on the ADOscore metamodel three services were identified - see figure 7: a scorecard service, a rating service, and a performance indicator service. The scorecard service implements the metamodel part in the form of a graphical model editor. It thus allows to create models of cause-effect relationships, e.g. as described above for the students' scorecards.

Next to this service, the rating service provides a visualization of the current quantification of the performance indicators assigned to the strategic goals in the cause-effect model. The data is thereby retrieved via the performance indicator service. This in turn accesses external data sources such as a personal database, e.g. regarding salary information, the university database, e.g. to assess the completion of the master curriculum, or a social network database, e.g. to determine the number of professional contacts.

With the resulting service oriented architecture all three components can now be easily modified independently of each other. For example, in case the cause-effect model in the scorecard service needs to be extended with additional concepts,

[4]http://www.boc-group.com/at/produkte/adoscore/

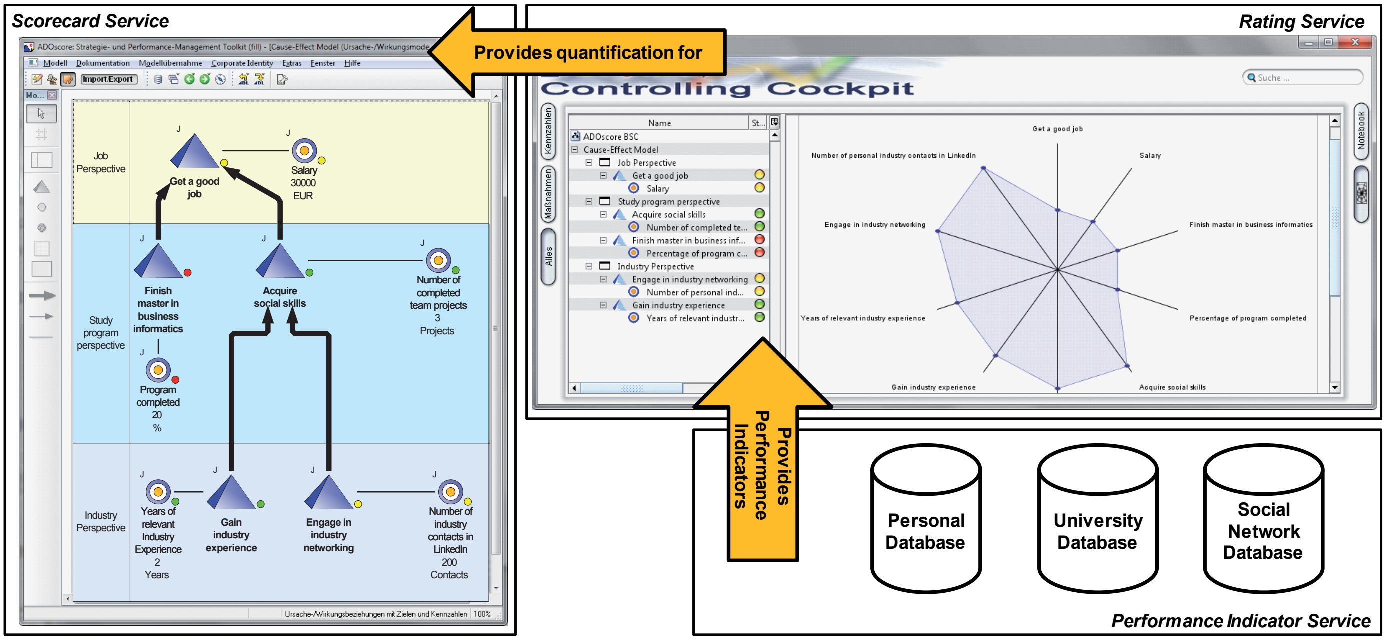

Fig. 7. Sample Instantiation of the Approach Using the ADOscore Strategy and Performance Management Toolkit

only the interface to the rating service has to be maintained. Similarly, if the data structure of the social network database changes, the performance indicator service is responsible for ensuring the alignment to the rating service. At the same time the underlying metamodel ties together all services not only on the technical level but also in terms of the according domain knowledge in order to optimally support requirements from a holistic performance management.

V. Evaluation

In order to give a first evaluation of the described approach and identify its limitations we discuss in the following its strengths, weaknesses, opportunities, and threats (SWOT).

A. Strengths

As already mentioned above the strengths of identifying services based on metamodels of enterprise modeling methods manifest themselves in the close alignment of business knowledge and technical data services. This leads to a service design that is inherently driven by business requirements, in contrast to approaches that would treat business requirements and technical implementations separately. The approach can thus be regarded as an extension of MDA concepts in software engineering.

B. Weaknesses

The major drawbacks of the approach are that the identification of possible services currently does not follow a formal methodology and that considerable business and technical knowledge is required. Although the use of model types in some enterprise modeling methods may facilitate the identification of potential services, it is hard to prescribe the best way of separating services and services in a way that is both technically feasible and makes sense from a business view.

Another limitation is the generalizability of the approach. In theory the approach is applicable to arbitrary metamodels. However, from a practical perspective additional entities may need to be added to existing metamodels, e.g. regarding information required for technical processing such as unique identifiers or security mechanisms.

C. Opportunities

With the transition from metamodels to technical services further possibilities for aligning business and IT perspectives emerge. For example, the further development of the approach could enable the creation of new business-driven IT applications on-demand by re-using services and model editors from different domains and for different application areas. This would be similar to the disussions around microservices, which are currently applied for modern cloud architectures[5]. However, the service data structure and interfaces would be more concrete based on the metamodel specifications than just by referring to business capabilities as it is done for microservices.

D. Threats

A potential threat of the approach poses the increased heterogenity of an organization's IT landscape. Despite the many advantages that are listed for service oriented architectures, large organizations may have difficulties in the future in keeping track of all their independent services and potentially incoherent implementations.

With the possibility of separating smaller and smaller parts of functionality in different blocks, also a number of organizational issues emerge. For example, the responsibility for ensuring the correct functioning of such services may be

[5]See http://martinfowler.com/articles/microservices.html accessed 05-08-2015

hard to track due to the numerous interactions with other services.

VI. Related Work

The concepts of decomposition and componentization as expressed by the metamodel slicing techniques have been used under different terms both in technical service-oriented contexts as well as in business contexts. We will therefore regard approaches in the field of software engineering as well as in the field of business transformation and enterprise architecture in the following.

From a software engineering perspective, several approaches have been discussed how models and modeling methods can be used in the context of service oriented architectures. In [9] an approach is presented that also relies on metamodels for defining the data that is to be exchanged between implementations of services. Thereby particular focus is put on how the transformations between different services specified through metamodels is accomplished. In [52] a modeling framework has recently been presented that permits to describe platform-independent and platform-specific models for supporting developers who need to interact with services. However, the main focus is thereby put on the technical implementation rather than the business content. Although these approaches in software engineering are considered to be very important for realizing technical implementations, the approach we described above differs from them in the way that it primarily considers the domain perspective rather than implementation aspects.

In the field of business transformation and enterprise architecture the approaches of pattern identification [53] and business componentization [11], [54] are related to our approach. In contrast to the metamodel slicing approach their contributions either take business decomposition or metamodelling patterns into consideration but without joining both results. Breaking it down to the information carrier, Dan et al. talk about information as a service (IaaS) to exploit the advantages of loose coupling and reusability [55].

In the context of metamodeling and services from a business perspective, the works of Braun et al. [56] and Winter et al. [10] are related to our approach. Although their approaches of extending the metamodels in business engineering with service concepts is similar, they do not include descriptions for the derivation of technical service implementations. On a technical level the service decomposition concept is treated similarly by Zimmermann et al. [57] and Quintero et al. [58]. Despite their similar approaches in regard to domain decomposition and service identfication, their focus is set on business process metamodels and not on arbitrary metamodels.

An approach that aims to join both business and technical views on services has very recently been proposed by Huergo et al. [59]. The MDCSIM method described therein reverts to master data and logical data models. Thereby, the master data stands for core enterprise information concepts and the logical data models are used to identify the relevant master data. By using a specific modeling technique, the business perspective is assessed to identify possible services. In comparison to our approach, MDSCIM does however not target arbitrary types of metamodels. This permits also to address non-data related aspects as well as knowledge concepts that are not directly represented by data [60], e.g. assumed cause and effect relationships of strategic goals.

VII. Conclusion

In this paper it has been shown how domain conceptualizations in the form of metamodels can be used to identify particular domain services. The proposed slicing techniques therefore do not only lead to a business driven service identification but may also be used to derive concrete technical service implementations. Further work on this approach includes the continued technical realization of the proposed concepts and the evaluation in a real life business setting. From a research perspective the described slicing techniques directly provide the basis for further work. This will include the analysis of methods and techniques to support and possibly automate the identification of slicing planes, e.g. based on artificial intelligence concepts and extensions of the semantic aspects of the metamodels [61], as well as the assistance through visualization techniques that are adequate in business settings.

References

[1] H. Chesbrough and J. Spohrer, "A research manifesto for services science," *Comm. of the ACM*, vol. 49, no. 7, pp. 35–40, 2006.

[2] R. Woitsch and D. Karagiannis, "Process-Oriented Knowledge Management Systems Based on KM-Services: The PROMOTE® Approach," *International Journal of Intelligent Systems in Accounting, Finance & Management*, vol. 11, pp. 253–267, 2003.

[3] G. Hui, "How the Internet of Things Changes Business Models," *Harvard Business Review*, vol. July 29, 2014.

[4] A. Musa, A. Gunasekaran, Y. Yusuf, and A. Abdelazim, "Embedded devices for supply chain applications: Towards hardware integration of disparate technologies," *Expert Systems with Applications*, vol. 41, pp. 137–155, 2014.

[5] H. Demirkan, R. J. Kauffman, J. A. Vayghan, H.-G. Fill, D. Karagiannis, and P. Maglio, "Service-oriented technology and management: Perspectives on research and practice for the coming decade," *ECRA Journal*, vol. 7, no. 4, pp. 356–376, 2008.

[6] D. Bork and H.-G. Fill, "Formal Aspects of Enterprise Modeling Methods: A Comparison Framework," in *Proceedings of the 2014 47th International Conference on System Sciences*. IEEE, 2014.

[7] M. Broy and I. H. Krueger, "A Formal Model of Services," *ACM Transactions Software Engineering and Methodology*, vol. 16, no. 1, pp. 1–40, 2007.

[8] I. Manolescu, M. Brambilla, S. Ceri, S. Comai, and P. Fraternali, "Model-Driven Design and Deployment of Service-Enabled Web Applications," *ACM Transactions on Internet Technology*, vol. 5, no. 3, pp. 439–479, 2005.

[9] A. Winter and A. Ebert, "Metamodel-driven Service Interoperability," in *STEP 2005 Workshop*. IEEE, 2005, pp. 147–158.

[10] R. Winter and J. Schelp, "Dienstorientierung im Business Engineering," *HMD*, vol. 241, pp. 45–54, 2005.

[11] M. Ernest and J. Nisavic, "Adding value to the IT organization with the Component Business Model," *IBM Systems Journal*, vol. 46, no. 3, pp. 387–403, 2007.

[12] C. Moser and F. Bayer, "IT Architecture Management: A Framework for IT Services," in *EMISA Workshop 2005*. Lecture Notes in Informatics, 2005.

[13] J. Schelp and R. Winter, "Towards a Methodology for Service Construction," in *HICSS'07 Conference*. IEEE, 2007, pp. 64–70.

[14] A. Erradi, S. Anand, and N. Kulkarni, "SOAF: An Architectural Framework for Service Definition and Realization," in *SCC'06 Conference*, 2006, pp. 151–158.

[15] D. Karagiannis, H.-G. Fill, P. Hoefferer, and M. Nemetz, "Metamodeling: Some Application Areas in Information Systems," in *UNISCON*, R. Kaschek and et al., Eds. Springer, 2008, pp. 175–188.

[16] G. Blair, N. Bencomo, and R. France, "Models@Run.Time," *IEEE Computer*, vol. 42, no. 10, pp. 22–27, 2009.

[17] H.-G. Fill and D. Karagiannis, "On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform," *Enterprise Modelling and Information Systems Architectures*, vol. 8, no. 1, pp. 4–25, 2013.

[18] H. Stachowiak, *General Model Theory (German)*. Springer, 1973.

[19] D. Karagiannis and H. Kuehn, "Metamodelling Platforms," in *Dexa 2002 Conference*. Springer Verlag, 2002, p. 182.

[20] J. Sprinkle, B. Rumpe, H. Vangheluwe, and G. Karsai, *Metamodelling - State of the Art and Research Challenges*. Springer, 2010, vol. LNCS 6100, pp. 57–76.

[21] D. Karagiannis and P. Hoefferer, *Metamodels in Action: An Overview*. Setbal: Insticc Press, 2006, pp. IS–27 – IS–36.

[22] H.-G. Fill, *Visualisation for Semantic Information Systems*. Gabler, 2009.

[23] H. Kern, A. Hummel, and S. Kuehne, *Towards a Comparative Analysis of Meta-Metamodels*. Portland, USA: ACM, 2011, pp. 7–12, http://www.dsmforum.org/events/DSM11/Papers/kern.pdf (last access 05-01-2012).

[24] K. Klose and R. Knackstedt, "Serviceidentifikation fuer die Produktionsplanung eines mittelstaendischen Auftragsfertigers (German)," *HMD*, vol. 253, pp. 47–56, 2007.

[25] M. P. Papazoglou, "Service-Oriented Computing: Concepts, Characteristics and Directions," in *WISE'03 Conference*. IEEE, 2003, pp. 3–12.

[26] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, and R. Metz, "OASIS Reference Model for Service Oriented Architecture 1.0," 2006, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm accessed 04-06-2007.

[27] H. M. Sneed and S. H. Sneed, "Creating Web Services from Legacy Host Programs," in *5th International Workshop on Web Site Evolution*. IEEE, 2003, p. pp. 59.

[28] F. Schoenthaler, G. Vossen, A. Oberweis, and T. Karle, *Business Processes for Business Communities - Modeling Languages, Methods, Tools*. Berlin-Heidelberg: Springer, 2012.

[29] H.-G. Fill, "Design of Semantic Information Systems using a Model-based Approach," in *AAAI Spring Symposium*. AAAI, 2009.

[30] ——, "On the Social Network based Semantic Annotation of Conceptual Models," in *7th International Conference on Knowledge Science, Engineering and Management*, R. Buchmann, C. Kifor, and J. Yu, Eds. Springer, 2014, pp. 138–149.

[31] ——, "An Approach for Analyzing the Effects of Risks on Business Processes Using Semantic Annotations," in *European Conference on Information Systems 2012*. AIS, 2012.

[32] H.-G. Fill, A. Gericke, D. Karagiannis, and R. Winter, "Modellierung fuer Integrated Enterprise Balancing," *Wirtschaftsinformatik / BISE*, vol. 06/2007, pp. 419–429, 2007.

[33] H.-G. Fill, "Using Obfuscating Transformations for Supporting the Sharing and Analysis of Conceptual Models," in *MKWI 2012*, S. Robra-Bissantz and D. Mattfeld, Eds. GITO Verlag, 2012.

[34] H.-G. Fill, T. Redmond, and D. Karagiannis, "FDMM: A Formalism for Describing ADOxx Meta Models and Models," in *Proceedings of ICEIS 2012, Wroclaw, Poland*, L. Maciaszek, A. Cuzzocrea, and J. Cordeiro, Eds., vol. 3, 2012, pp. 133–144.

[35] H.-G. Fill, "On the Conceptualization of a Modeling Language for Semantic Model Annotations," in *Advanced Information Systems Engineering Workshops, CAiSE 2011*, C. Salinesi and O. Pastor, Eds., vol. LNBIP Vol. 83. London, UK: Springer, 2011, pp. 134–148.

[36] A. Sheth, K. Verma, and K. Gomadam, "Semantics to energize the full services spectrum," *Comm. of the ACM*, vol. 49, no. 7, pp. 55–61, 2006.

[37] K. Hinkelmann, S. Nikles, and L. Von Arx, "An Ontology -based Modelling Tool for Knowledge-intensive Services," in *SBPM 2007 Workshop*, M. Hepp, K. Hinkelmann, D. Karagiannis, R. Klein, and N. Stojanovic, Eds., 2007.

[38] J. Cardoso, "Discovering Semantic Web services with and without a Common Ontology Commitment," in *SCW'06 Workshop*, 2006, pp. 183–190.

[39] H.-G. Fill, J. Lischka, and D. Karagiannis, *Web-Services und Geschaeftsprozesse im Semantic Web*. Berlin et al.: Springer, 2006, pp. 321–335.

[40] H.-G. Fill and P. Burzynski, "Integrating Ontology Models and Conceptual Models using a Meta Modeling Approach," in *11th International Protege Conference*, 2009.

[41] C. Lichka, "Business Scorecarding as model-driven approach for strategy implementation," PhD thesis, University of Vienna, Austria, 2006.

[42] ——, "Strategic Monitoring and Alignment to Achieve Business Process Best Practices," in *Database and Expert Systems Applications Workshop*. IEEE, 2005, pp. 914–918.

[43] W. C. Kim and R. Mauborgne, "Blue Ocean Strategy," *Harvard Business Review*, no. October 2004, pp. 76–95, 2004.

[44] R. Nolan and W. McFarlan, "Information Technology and the Board of Directors," *Harvard Business Review*, no. October 2005, pp. 96–106, 2005.

[45] H.-G. Fill, "Using Semantically Annotated Models for Supporting Business Process Benchmarking," in *10th International Conference on Perspectives in Business Informatics Research*, J. Grabis and M. Kirikova, Eds. Springer, 2011, pp. 29–43.

[46] R. Kaplan and D. Norton, "Using the Balanced Scorecard as a Strategic Management System," *Harvard Business Review*, vol. January-February 1996, pp. 75–85, 1996.

[47] D. Karagiannis, F. Ronaghi, and H.-G. Fill, "Business-oriented IT management: developing e-business applications with E-BPMS," in *ICEC 2008*, vol. Vol. 258. ACM, 2007, pp. 97–100.

[48] P. Fettke and P. Loos, "Model Driven Architecture (MDA)," *BISE*, vol. 45, no. 5, pp. 555–559, 2003.

[49] N. Visic, H.-G. Fill, R. Buchmann, and D. Karagiannis, "A Domain-specific Language for Modeling Method Definition: from Requirements to Grammar," in *RCIS'2015*. IEEE, 2015.

[50] P. Rittgen, *Handbook of Ontologies for Business Interaction*. IGI Global Press, 2008.

[51] W. Brenner, D. Karagiannis, L. Kolbe, H. Krueger, H.-J. Lamberti, L. Leifer, J. Leimeister, H. Oesterle, C. Petrie, H. Plattner, G. Schwabe, F. Uebernickel, R. Winter, and R. Zarnekow, "User, Use and Utility Research : The digital user as new design perspective in Business and Information Systems Engineering," *Business and Information Systems Engineering*, vol. 6, no. 1, pp. 55–61, 2014.

[52] C. Atkinson, P. Bostan, and D. Draheim, *A Unified Conceptual Framework for Service-Oriented Computing*. Springer, 2012, pp. 128–169.

[53] H. Kuehn, "Methodenintegration im Business Engineering," PhD thesis, University of Vienna, 2004.

[54] Sanz, J. L. C. et al., "Business Services and Business Componentization: New Gaps between Business and IT," in *IEEE SOCA '07*, 2007, pp. 271–278.

[55] A. Dan, R. Johnson, and A. Arsanjani, "Information as a Service: Modeling and Realization," in *Proceedings of the International Workshop on Systems Development in SOA Environments*, 2007, pp. 2–7.

[56] C. Braun and R. Winter, "Integration of IT service management into enterprise architecture," in *Proceedings of the 2007 ACM symposium on Applied computing*. ACM, 2007, pp. 1215–1219.

[57] O. Zimmermann, N. Schlimm, G. Waller, and M. Pestel, "Analysis and Design Techniques for Service-Oriented Development and Integration," in *INFORMATIK 2005*, vol. 68. GI, 2005, pp. 606–611.

[58] R. Qunitero, V. Torres, M. Ruiz, and V. Pelechano, "A Conceptual Modeling Approach for the Design of Web Applications based on Services," in *Proceedings of the 44th annual Southeast regional conference*. ACM, 2006, pp. 464–469.

[59] R. Huergo, P. Pires, and F. Delicato, "MDCSIM: A method and a tool to identify services," *IT Convergence Practice*, vol. 2, no. 4, pp. 1–27, 2014.

[60] T. Le Dinh, T. Rickenberg, H.-G. Fill, and M. Breitner, "Enterprise Content Management Systems as a Knowledge Infrastructure: The Knowledge-Based Content Management Framework," *International Journal of e-Collaboration*, vol. 11, no. 3, pp. 49–70, 2015.

[61] A. Leutgeb, W. Utz, R. Woitsch, and H.-G. Fill, "Adaptive Processes in E-Government - A Field Report about Semantic-based Approaches from the EU-Project FIT," in *Proceedings of the International Conference on Enterprise Information Systems (ICEIS 07)*. INSTICC, 2007, pp. 264–269.