Jörg Becker, Daniel Beverungen, Ralf Knackstedt, Oliver Müller

Model-Based Decision Support for the Customer-Specific Configuration of Value Bundles

Customers in B2B as well as B2C markets increasingly demand integrated problem solutions from their suppliers, comprising both physical artefacts (products) and services. Applying a mixed bundling strategy to offer such value bundles to customers foremost requires a sound configuration and economic calculation of value propositions, based on previously defined modules of products and services. In this paper, a modelling language is introduced to describe the function and structure of such modules, as well as to calculate the economic consequences of value propositions on a customer-individual level. The proposed modelling language has been embedded into a software tool to evaluate its utility regarding the customization and offering of integrated value bundles to customers.

1 Introduction

According to a service-dominant logic [VaLu04], [VaLu07] customers increasingly demand integrated problem solutions that fit their individual needs instead of requesting standardized physical products. One way for suppliers to satisfy this demand is to offer integrated value bundles—consisting of physical products and related services [HaKo07]—as value propositions for customers. Value bundles comprise separately marketable products and services. They can be offered as individual value propositions for customers. If the value proposition is accepted by customers, value bundles are delivered in a service process that needs to be integrated into the customer's processes and therefore requires customer input. Outcomes for customers to be gained from value bundles can have tangible and intangible aspects. As a result of this integration, value bundles are able to create outcomes for customers higher than the summed-up outcomes of their components [Schm08]. Already today, offering pure physical products is seldom, as can be comprehended when looking at the services provided in retail. Thus, distinguishing products from services becomes increasingly challenging [FiFi01], [Tebo06], [VaLu07].

The increasing dominance of the service sector [OECD05] further amplifies the need for suppliers to develop and provide integrated value bundles as problem solutions for their customers. This is especially true for the German Mechanical Engineering and Electrical Engineering industries. Evaluating results from two broad empirical studies in both sectors, Stille comes to the conclusion that turnover related to services has doubled in the Electrical Engineering sector from 9.6% (1997) to 18.5% (2000) and significant gains from 16.8% (1997) to 22.5% (2000) could be identified in the Mechanical Engineering Sector [Stil03]. Moreover, Mercer Management Consulting points out, that half of the growth in German Mechanical Engineering in the years 1998-2003 can be accounted to exploiting the potential of services. Likewise, the margin of the service business (10%) is significantly higher than the margin of the product business (2.3%). Furthermore, they state that margins could be even higher when looking at some leading edge services only, which catch margins up to 18% [Merc03]. Additional empirical research shows that companies attribute a high (38.1%) or very high (59.8%) impact on their revenues to their service business. Services are also seen as a good means for differentiation from competitors as well as for superior customer retention. Consistently, 94.9% of the companies examined have plans to grow their business by offering value bundles [StBa+07].

While the necessity to offer services is widely acknowledged, manufacturing companies articulate severe difficulties to systematically describe their service portfolio. Such difficulties seem plausible when considering the apparent lack of modelling approaches for formally describing value bundles [BeBe+08]. Creating physical products according to formalized specifications has long been in focus of the engineering disciplines and has lead to a considerable degree of standardization. Especially STEP (ISO 10303-41: Fundaments of Product Description and Support; ISO 10303-42: Geometric and Topological Representation; ISO 10303-46: Visual Presentation) has gained particular importance in *Product Engineering* [AnTr00]. Drawing from experiences from product engineering, adapting traditional engineering techniques to the design of services has been discussed under the label *Service Engineering* since the 1990s [Ganz06], with a focus in governmentally funded service research in Germany. Since then, a number of modelling languages for engineering services have been proposed (e.g. [CoGö03], [Klei07], [KuLo+05], [Lucz91], [Shos77], [Shos82], [WiLu06]; a more exhaustive overview is provided by [BeBe+08], as well as [Emmr05]). However, a consolidation of approaches similar to the standardization efforts in product engineering cannot be ascertained. Hence, modelling of value bundles can be seen as a next step of evolution, integrating approaches from both disciplines, product engineering and service engineering. Some approaches have recently been proposed (e.g. [Bott07], [Emmr05], [More02], [ScGr+06]), although they have not been evaluated or established in real-life scenarios.

This paper advances the discussion on developing modelling languages for value bundles. Following Alexander's advice to decompose design problems (in our case designing tailored value bundles) in terms of function on the one hand and economics on the other hand [Alex70], our modelling language specifically addresses two major challenges: First, each value bundle offered as a value proposition for a particular customer is contingent on his preferences, i.e., needs, wants and demands [Arnd78]. One option to cope with this variety and reap economies of substitution [GaKu03] is to follow a modularization approach by 'assembling' individual value bundles from an array of pre-defined product and service modules. To account for this *configuration*, a suitable modelling language must be able to represent the possible solution space (consisting of atomic elements—i.e. products and services—and their attributes, modules, structures, and configuration rules) of value bundles for suppliers and customers in an appropriate manner. Second, a suitable modelling language shall support *calculation* in terms of economic consequences imposed on suppliers and customers when selecting different configurations of value bundles. Thus, *decision support* for selecting an appropriate value bundle is conveyed. As the complexity caused by these two challenges requires the modelling language to be implemented into a suitable software tool, we developed a first prototype.

The remainder of the paper is organized as follows: First, we briefly introduce the specific requirements of modelling value bundles that are addressed by our approach (Section 2). Building on these premises, we show the construction of the modelling language (Section 3) and introduce a first software prototype to support the modelling, configuration and calculation of value bundles (Section 4). We close with a brief summary and show perspectives for further research (Section 5).

2 Requirements for modelling value bundles

In general, requirements put towards a modelling language originate from the contexts the language will be used in. Due to the broad array of contexts modelling languages for value bundles might be used in—ranging from price calculations to the integration of related business processes in inter-organizational networks—requirements can be quite diverse. A classification of requirements might be accomplished by adhering to the views of service potential, service process, and service result [Hilk89], as commonly used in service science. From a (service) *potential* point of view, a suitable modelling language must support the modelling of resources required to provide services and manufacture products [ScGr+06]. Additionally, modelling the capacity of these resources is fundamental. From a (service) *process* point of view, processes to perform service and manufacturing activities are to be represented and integrated with each other to account for the value bundle to be an integrated, customer-specific solution. In this context, one major aspect is to account for the involvement of the customer as a co-creator of value [VaLu04], [VaLu07]. From a (service) *result* point of view, the structure and properties of the value bundle and its value proposition for the customer has to be modelled.

Additional requirements arise from the intended adoption of a modularization strategy [Boeh04]. To apply this strategy it must be possible to compose value bundles from previously defined modules. This mass-customization approach may allow for exploiting economies of substitution [GaKu03]. Benefits include re-using existing knowledge associated with product modules and service modules, reducing performance slippage when incorporating additional modules into the bundle, reducing incorporation costs for suppliers and customers and—maybe most important—making value bundles modularly upgradeable to cope with changing customer preferences [BaCl97], [GaKu03]. A prerequisite to assemble bundles from modules are hierarchical structures (i.e. is-part-of relationships) of modules as well as non-hierarchical relationships (i.e. configuration rules, e.g., condition, exclusion, substitution, enhancement) between modules.

	Supplier's point of view	Customer's point of view
Representation of function and structure	(I) Modelling possible configurations of value bundles; i.e. modelling of the solution space.	(II) Configuration of customer-specific value bundles; i.e. deriving customerspecific instances from the solution space.
Representation of economic consequences	(III) Calculation of costs related to the acquisition and application of ressources required to provide and combine products and services; i.e. economic consequences for the supplier.	(IV) Calculation of e.g. the total cost of ownership (TCO) of a specific value bundle; i.e. economic consequences for the customer.

Figure 1: Requirements towards modelling languages for value bundles

Requirements also originate from the different points of view which suppliers and customers have on value bundles. From the supplier's point of view, modelling a solution space of consistent (i.e. buildable and desirable) configurations is essential. To make the definition of such a solution space possible, products and services have to be described adequately and in a formalized notation. Additionally, configuration rules need to be specified to ensure that the customer-specific value bundles to be configured will be consistent. From the customer's point of view, the modelling language must support the derivation of individually customized value bundles subject to the previously defined generic solution space. This configuration should be made according to the preferences of the particular customer to derive a portfolio of value bundle instances fitting customer's needs, wants, and demands.

For a sound selection of value bundles, the economic consequences of a decision—for both suppliers and customers—have to be taken into account. For suppliers, it is e.g. crucial to calculate the capital value of different value bundles as well as the capital value of single modules. From the customer's point of view, e.g., original and derivative payments along the entire lifecycle (i.e. total cost of ownership) of the value bundle are particularly relevant.

Figure 1 summarizes and classifies the stated requirements using functional and economic criteria on the one hand and the supplier and customer perspective on the other hand. In the following sections we design and apply a modelling language for value bundles addressing these requirements, with a focus on quadrants I, II and IV.

3 A modelling language for value bundles

Modelling languages are essential for building models [Schu98]. Generally, a modelling language comprises a conceptual language aspect and a representational aspect [Holt00]. The conceptual language aspect defines the meaning of the modelling constructs and relationships among them. The representational aspect assigns a notation of symbols to these constructs to make them easier to grasp and use for developers and users. The conceptual language aspect of the proposed modelling language is depicted in the Entity Relationship Diagram in Figure 2. When creating a specific model the presented language constructs will be instantiated. While most constructs (e.g. outcome, resource) have a graphical representation in the modelling tool, some are only visible via a context menu (e.g. value, operator, unit of measurement). Figure 4 summarizes the representational language aspect by presenting some exemplary models.

The starting point for modelling value bundles is the construct *Value Bundle (type)*. It represents the solution space, i.e., all valid configurations of a type of value bundle (e.g. a machine centre with related services) as seen from the supplier's point-of-view (see also the concept of the "generic product model" proposed by [Sche06]). It comprises the hierarchical structure of the bundle, available modules consisting of product and service elements, attributes of atomic product and service elements as well as configuration rules. The Value Bundle (type) construct hence can be regarded as a knowledge base capturing both product and service information and spans the solution space from which customer-specific bundles can be derived.

The static aspect of a value bundle on type level is primarily defined by the modules it is composed of. *Modules* are self-contained units or building blocks containing outcomes (i.e. products or services), which can be re-used in different value bundles. Hence, besides defining bundles from scratch, it is also possible to create models of value bundles (type) by combining existing building blocks, which speeds up the modelling process. This can be referred to as a mixed bundling approach [HaKo07].

As already stated, modules are themselves composed of *Outcomes*. Outcomes are the result of some economic factor combination and can be products or services. As the differentiation of products and services becomes increasingly problematic [SaFr06], [Tebo06], [VaLu04], [VaLu07], we refrain from making an explicit distinction. Especially in industrial contexts, such as the mechanical or electrical engineering industry, service processes regularly involve physical components, e.g., spare parts to be used during their execution or vice versa. When modelling with the proposed language, it has to be ensured, that all outcomes associated with the same module provide a similar value function for the customer. They have to be perceived as more or less interchangeable value propositions that can be selected from by the customer during the configuration process. Hence, outcomes often vary only in terms of their non-functional attributes, such as quality, quantity or price. Outcomes can be arranged in *Outcome Hierarchies*, i.e., outcomes can again be composed of other outcomes. On the one hand, this enables the representation of hierarchical structures as regularly used to describe physical products (e.g. bill of materials). On the other hand, activities found in service processes can be modelled as hierarchical sequences (e.g. maintenance as a sequence of 'analyze error', 'resolve error' and 'verify resolution'). This hierarchical organization of components is a common means to foster the description and reuse of components [GaKu03].

Outcomes are described by *Attributes*. When describing rather physical outcomes, widely used physical (e.g. dimensions), mechanical (e.g. revolutions per minute) or technical (e.g. clock rate) attributes might be applied. As services can be highly intangible and heterogeneous such attributes may not be suitable to describe outcomes which are rather pure services (e.g. consulting services). Instead, functional and non-functional attributes should be used. Functional attributes describe the result of the service process as perceived by the customer. Depending on the type of involvement of the customer during the service process (involvement of the customer himself or herself, involvement of an object of the customer, provision of information [PaSt+08]) functional attributes may vary. Non-functional attributes represent constraints or conditions referring to the provided function. Examples can be as diverse as price, quality, quantity, availability, or delivery and payment conditions [Osul06]. In addition, attributes referencing to standard classification systems (e.g. UNSPSC, NAICS, eCl@ss) are useful to search for, compare and select different outcomes.

Outcomes are further described by *Preferences* and *Preference Values* to support the customer during the configuration process. By asking the customer for the relative importance of each assigned preference (e.g. availability) and the desired degree of preference fulfilment (e.g. 99%) it is possible to provide further decision support, e.g., by calculating preference scores for each outcome and ranking outcomes respectively.

To restrict the solution space from which customer-specific value bundles can be derived, to guide the configuration process and to assure consistency the construct *Configuration Rule* is used. The proposed modelling language provides such rules in if-then form. Configuration rules can be used on a module level as well as on an outcome level. If a configuration rule refers to a whole module, an outcome and an attribute have to be specified for the if-part and the then-part respectively. In case a configuration rule refers to a specific outcome, this is unnecessary. In both cases, operators (<, <=, =>, >, =, !=) must be specified to clarify the mode of operation (enhancements of the configuration rule construct in accordance to e.g. the OMG Specification Production Rule Representation [OMG07] are planned).

As value bundles regularly comprise physical products and related services along their lifecycle, models of value bundles have to account for this dynamic aspect. This is achieved by introducing the constructs *Lifecycle Phase* and *Interval* to model different stages (e.g. pre-sales, operation, and end-of-life) and time periods (e.g. year 1, year 2, year 3) of lifecycles. Both, value bundles and single outcomes can be subject to a lifecycle. For example, in the value bundle "financed, maintained and sustainable machine centre" financing services may be provided in the pre-sales phase, the machine centre itself and maintenance services in the operation phase, and recycling services at the end of life. All these outcomes again have a distinct lifecycle; e.g. for the machine centre: design, construction, assembly.

Intervals are used to detail lifecycle phases. An interval may comprise several *Activities* to be carried out. As opposed to outcomes, activities are units which are not marketed separately, e.g., particular work-steps. An activity can be carried out by a *Business Unit*, which might be further subdivided into *Jobs*. More-over, activities comprise operand (the resources to be worked on, e.g., raw materials, additives or supply items) and operant (the resources used to transform the resources to be worked on, e.g., machinery, information and skills) *Resources* [VaLu04]. By specifying resources and jobs, the potentials needed for offering value propositions to customers can be modelled. For the purpose of economic evaluations, both constructs can be assigned to *Cost Centres*. For convenience, nevertheless it is possible to assign business units and resources directly to intervals, with no need to specify detailed activities.

In the context of services the construct *Customer Resource* is required to account for the integration of the customer as a co-creator of value into service processes. Using this construct it is possible to specify which input the customer has to provide for a service process to be carried out effectively and efficiently [PaSt+08]. Typical customer resources are information (e.g. problem descriptions), employees (e.g. operating personnel), objects (e.g. a machine to be maintained) or rights (e.g. the right to shut down an assembly line).

During the process of configuring individual value bundle instances, the customer selects outcomes as modelled by the supplier by matching the attributes of outcomes with his or her own preferences. The modelling tool supports this process by presenting the options to be selected from (i.e. modules and outcomes) and by assuring compliance with the underlying configuration rules. The result of the configuration process is represented in the relationship type *Configuration* in Figure 2. The set of these relations for one value bundle on type level defines the configuration of a customer specific value bundle on instance level and forms the basis for the calculation of its economic consequences. From the customer's point of view, the configured value bundle represents a variant of the generic value bundle described by the supplier.

4 Tool-support for modelling, configuring, and calculating value bundles

The presented modelling language has been implemented into a meta-modelling software tool. The client-server architecture of the tool allows for distributed modelling, configuration and calculation of value bundles. It supports the modelling of generic value bundles via standalone-clients as seen from a supplier's point of view, and the configuration of customer-specific value bundle instances as well as the calculation of the economic consequences of selecting a specific value bundle instance via web-clients. Thus, suppliers and customers are provided with decision support on which value bundles to sell or buy, respectively. In the following sections the tool support of this overall process of modelling, configuration, and calculation is described (cf. Figure 3).

4.1 Modelling value bundle types

The process starts with the supplier modelling possible configurations of generic value bundles (solution space) that determine which specific variants are generally feasible (cf. quadrant (I) of Figure 1). Typically, suppliers model value bundle types in a bottom-up fashion (cf. Figure 4). First, internal and external as well as customer resources required for the provisioning of an outcome are specified. These resources are then assigned to activities, organizational units, and cost centres and described by attributes (e.g. costs per unit, consumption, duration). Second, lifecycle structures, outcome attributes, preferences and preference values are defined. Third, outcomes are composed of the entities modelled in steps one and two. Fourth, value bundles are defined by assigning outcomes to modules and specifying the value bundle lifecycle. As integrated value bundles are seldom provided by a single supplier [BaRo08] the tool allows for the distributed modelling of all the entities described above.

4.2 Configuring value bundle instances

Customers can configure individual value bundle instances via a web-based configurator (cf. quadrant (II) of Figure 1).

The configuration process can be conducted in several ways: (1) by the customer himself/herself, (2) by a key-account manager of the vendor who assembles a value bundle for a customer in the back-stage, or (3) the value bundle is jointly configured in a consulting process with representatives from both parties. Thus, different organizational setting can be realized to fit a variety of scenarios. For example, customers might fail to configure very complex solutions of products and services on their own due to a lack of knowledge and need the advice of a sales representative. On the other hand, in case the products and services are rather standardized and require little consulting during the process of configuration, vendors might want to outsource the process of configuration to their customers. In this scenario, customers can configure their value bundle as they would in an usual E-Commerce setting.

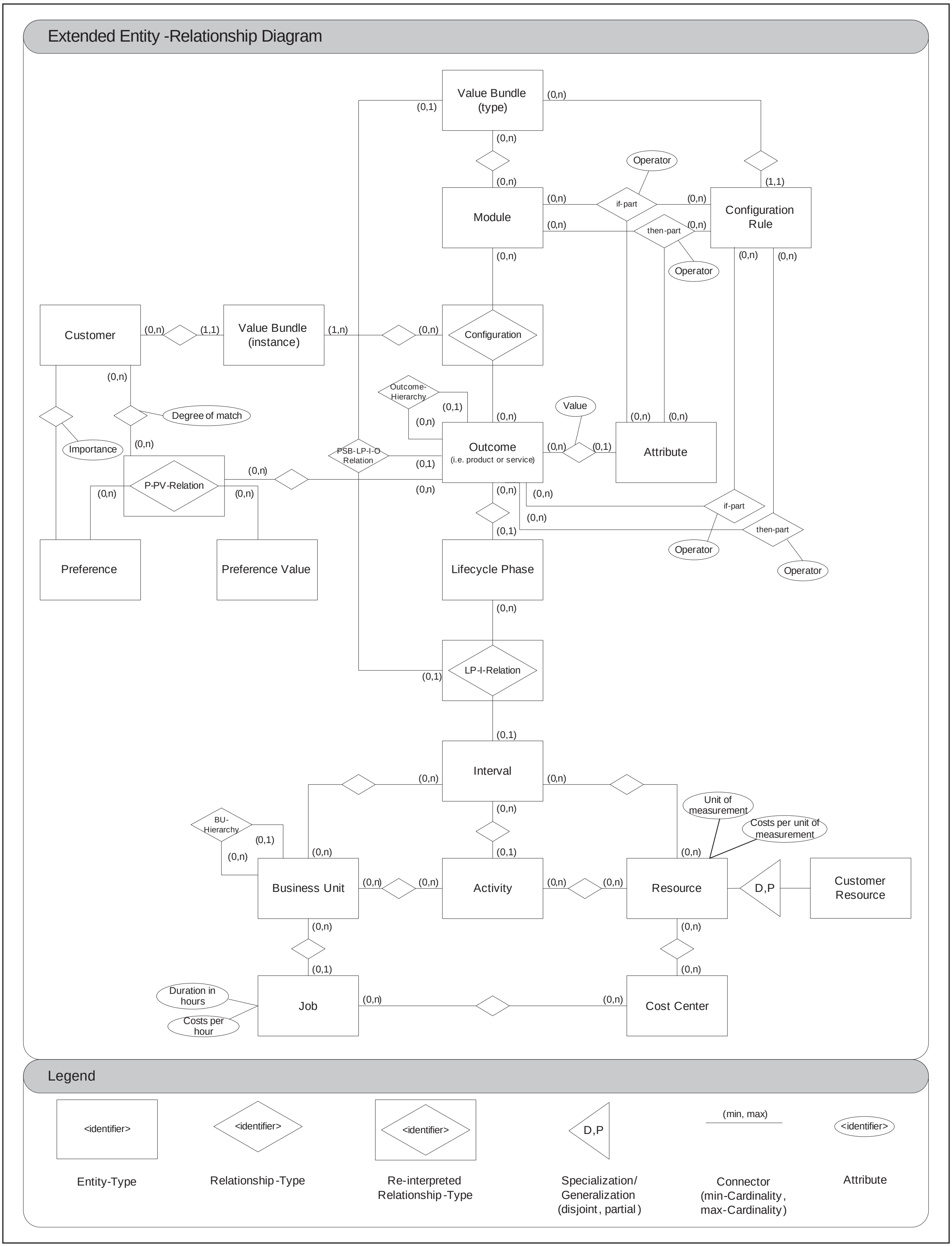

Figure 2: Meta-model of the proposed modelling language

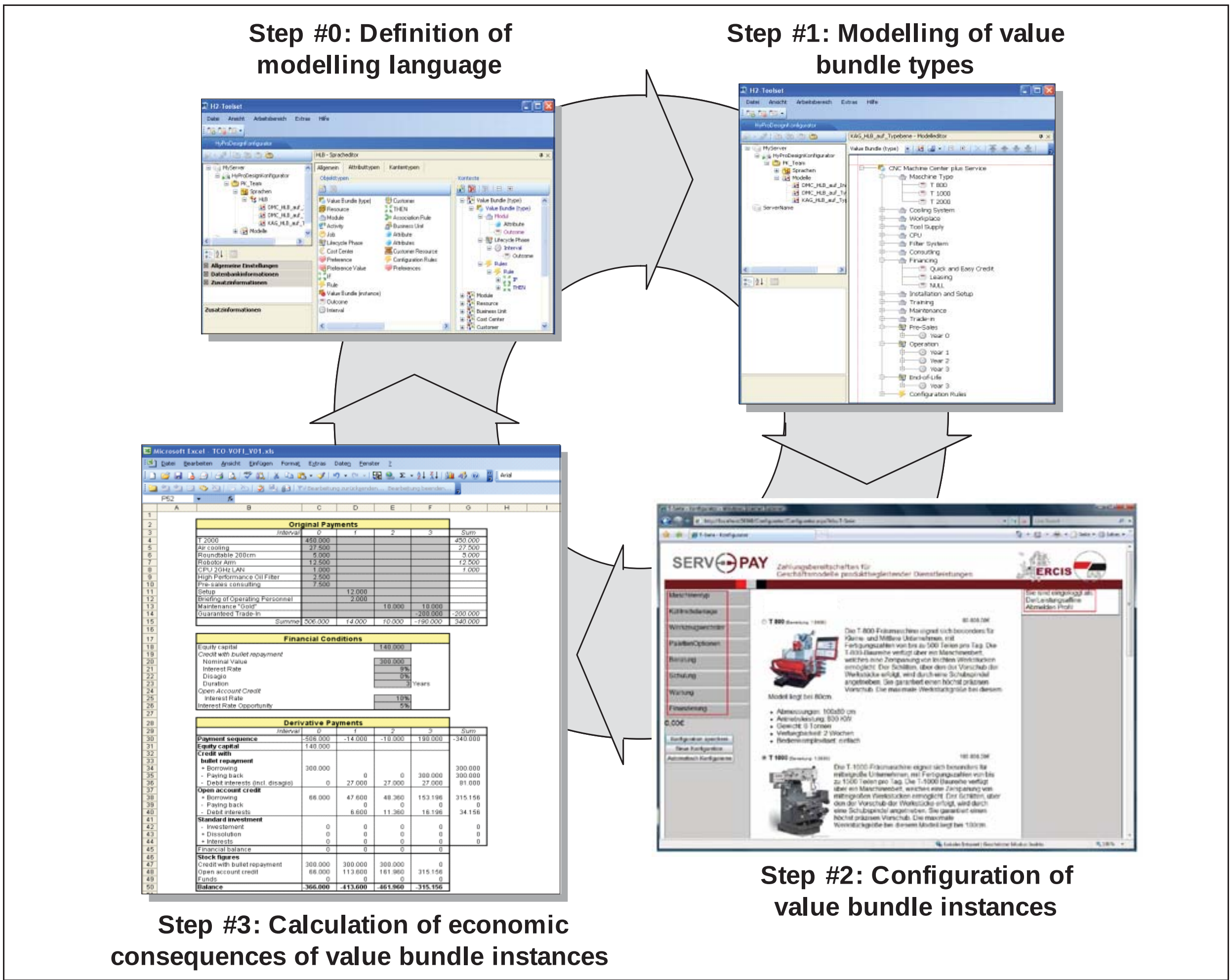

Figure 3: Modelling, configuration and calculation of value bundles with the proposed modelling language

In either way, before starting the configuration the customer has the possibility to state a number of preferences (e.g. price, availability, delivery time) and assign relative weights to each preference. Based on this information the configurator is able to make recommendations of suitable outcomes for each module or can even auto-configure whole value bundles.

The score of a preference is defined by two components: the importance of the preference (i.e. imp_A) and the degree of fulfilment of the linguistic preference value (i.e. p_A) [ArGo00], both of which are to be specified by the user before the configuration process takes place.

Preference scores are calculated using the fuzzy AND formula (cf. Formulas below), as proposed by [ArGo00]. The calculation of a preference score is based on the 'matching degree' of the preference as stated by the customer and the preference-value-outcome assignments made by the supplier (cf. Figure 4). The value of this match is in [0, 1] with 1 being a perfect match, whereas 0 represents no match.

The importance of a preference describes how strongly the preference influences the customer's decision. For example, the *delivery time* of a value bundle might have the importance of imp_A = 0.85. Consecutively, for preference p_A the customer has to judge in how far the stated preference value satisfies his desired degree of fulfilment. For example, the customer can articulate that a fast delivery time fulfils the preference *delivery time* with a value of 0.6, whereas a medium delivery time fulfils with 0.3 and a long delivery time with only 0.1.

Next, for all preferences associated to an outcome the individual preference score is computed according to formula (I). In our example this would result in the scores 0.66 for a fast delivery time, 0.4 for a medium delivery time and 0.24 for a long delivery time. In a second step, the scores of all preferences associated

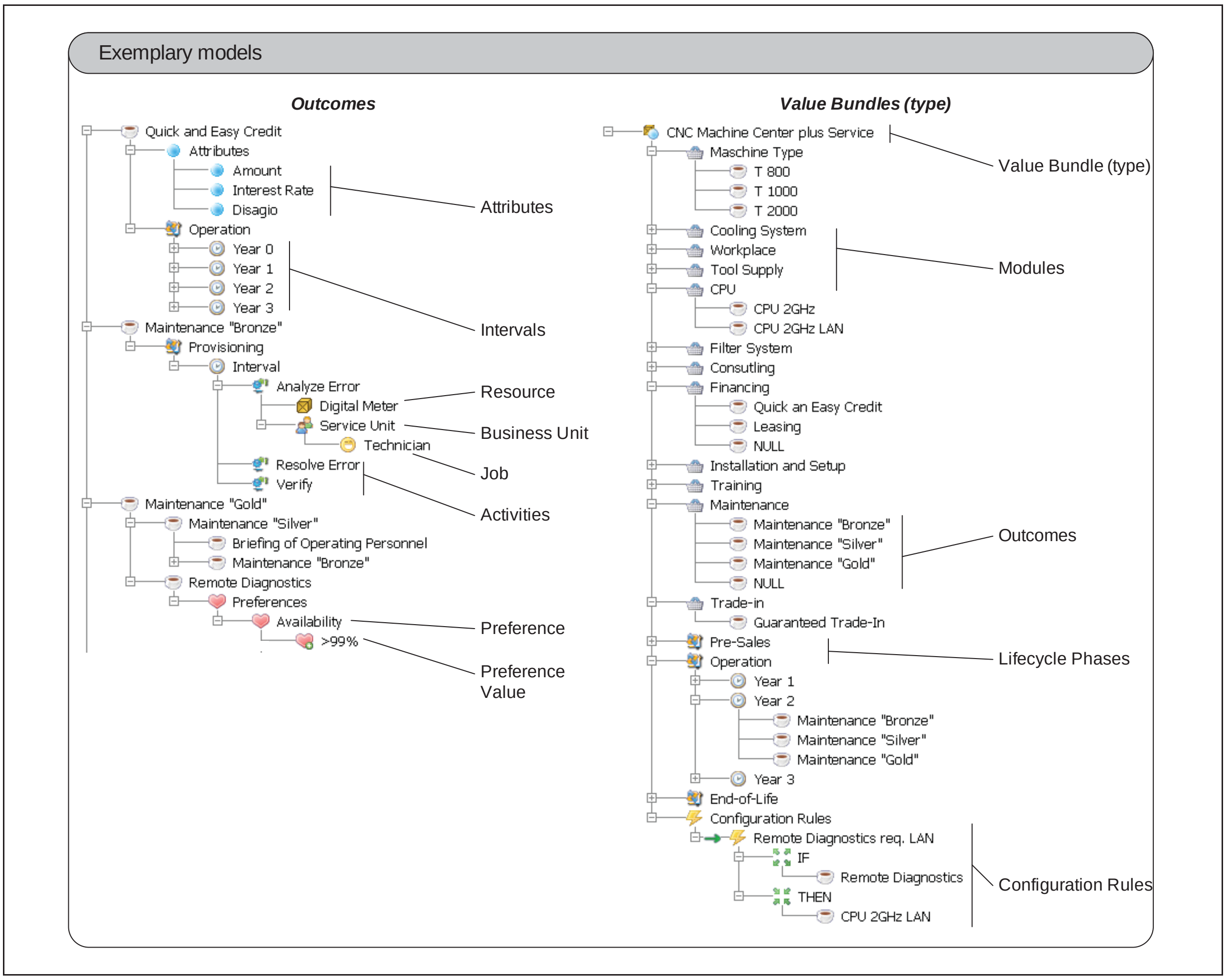

Figure 4: Notation and exemplary models of outcomes and value bundles

to an outcome are aggregated to an overall score by utilizing formula (II) in the case of exactly two preferences, or formula (III) in the case of more than two preferences.

$$score_A = imp_A \cdot p_A + (1 - imp_A) \qquad \text{(I)}$$

$$SCORE(score_A, score_B) = \frac{score_A \cdot score_B}{score_A + score_B - score_A \cdot score_B} \qquad \text{(II)}$$

$$SCORE(score_A, score_B, score_C) = \frac{SCORE(score_A, score_B) \cdot score_C}{SCORE(score_A, score_B) + score_C - SCORE(score_A, score_B) \cdot score_C} \qquad \text{(III)}$$

The results are presented to the customer by ranking the outcomes of a module according to their overall preference score. A threshold can be defined to automatically discard outcomes with low scores. Optionally, it is even possible to auto-configure the whole value bundle by automatically selecting the highest ranked outcome for each module. The customer, key-account manager or both can then browse and analyse the resulting value bundle instance and add or remove specific outcomes from the selection manually.

All configurations made by customers are stored in the central repository of the modelling tool for further analysis. One example for such analyses is a cluster analysis which can be used to identify recurring patterns and find promising bundles to be promoted in consecutive configurations. Another possibility is the generation of association rules which can be used to make recommendations based on historic sales data. These recommendations can be used to propose configurations, such as: “customer who bought the T-500 machine also decided for a 10 year all-inclusive warranty". Also, aborted configuration processes are recorded and stored to document the point at which the configuration was cancelled by the user. This information can be used to detect a lack of software usability.

4.3 Calculating economic consequences

As the process of configuration might not always lead to a clear decision right away but instead recommend a range of different configurations, the calculation of the economic consequences of each configuration might provide further decision support (cf. quadrant (IV) of Figure 1).

Due to the longevity bundles, their sound selection is comparable with traditional capital investment decisions. Therefore, methods used in investment controlling—which also consider long-term economic consequences of decisions—are suitable for an evaluation of value bundles. An investment appraisal typically considers the pay-ins and pay-outs as well as the available equity capital to apply financial calculations [Grob89]. From this payment sequence, derivative payments (e.g. cost of capital or tax) can be derived and allow for the computation of the final value of an investment. Beyond that, the total cost of ownership can be calculated if all original and derivative payments along the lifecycle of the value bundle, as well as imputed interest rates for the deployed equity capital are considered. The resulting value can be interpreted as the total cost of ownership (TCO) of the value bundle [GrLa04], [Vmb06].

For each value bundle configured from the generic solution space, the payment sequence according to the value bundle lifecycle, the selected modules and outcomes can be built. This is done by applying an algorithm similar to the pseudo-code fragment depicted in Figure 5.

Crucial for computing the overall payment sequence of a value bundle is to correctly assign the payments of the individual outcomes to the correct period of the overall value bundle payment sequence, as both periods might be dissimilar.

Payments made by customers at the time of sale are one-time payments to be assigned to period 0 of the overall payment sequence. Consecutive payments associated with an outcome might have to be split and reassigned according to the amount and length of the periods of the overall payment sequence. Once this task has been performed, payments are stored in a payment sequence array. Empty entries are filled up with zeros. Consecutively, the payment sequence array is further processed to (1) display the payment sequence in an HTML-Table which is displayed if a mouse-over event related to an outcome is detected

```
Establish connection to the configured model;
Read value bundle from model and save selected modules
     according to lifecycle stages (e.g. Sales, After-Sales
     and Recycling) into ArrayLists. Only stages which imply
     payments to be made by customers shall be included;
For each lifecycle stage
{
    Compute payments associated with each value bundle in
        this stage
    If (lifecycle stage == "Sales") payments are one-time
        payments and are assigned to the first period only;
    If (lifecycle stage is longer than period length)
        payments need to be partitioned to fit into several
        consecutive periods;
    If (lifecycle stage is shorter or equals period length)
        payments have to be associated with one period only;

    in either way for each of the selected periods
    {
        Set payments of preceding (unutilized) periods = 0;
        Calculate payments per day for each of the selected
            value bundle items, by computing (average
            payments per day + additional payments per day)
            * noOfDaysInSelectedPeriod;
        Set payments in unutilized consecutive periods = 0;
        Update computed values in the ArrayLists;
    }
}
Display the payment sequence
    In the web-based configuration and calculation tool by
        generating an HTML Table;
    Export payment sequence into an Excel sheet to compute
        Total Costs of Ownership according to additional
        factors, such as interest rates, equity capital or
        other factors influencing TCO as perceived by the
        customer;
```

Figure 5: Pseudo-code fragment for computing the payment sequence of a value bundle instance

in the configuration tool and (2) to export the payment sequence into an MS Excel template sheet for further analysis. In this pre-defined sheet, additional customer-specific variables, esp. available equity capital and loans incl. interest rates and duration, can be adjusted by the customer to calculate the expected costs of ownership for the desired configuration from his or her own point of view.

In the following, the cost calculation is illustrated by means of calculating an exemplified use case (cf. Figure 6). ACME Corporation has configured a value bundle consisting of a CNC Machine Centre and related services. The physical items to be capitalized amount to 450,000 EUR for the basic module, 27,500 EUR for the Air Cooling, 5,000 EUR for a Roundtable, 12,500 EUR for a Robot Arm, 1,000 for a CPU, and 2,500 EUR for an Oil Filter; all due in Interval 0. The desired services comprise pre-sales consulting (7,500 EUR) in Interval 0, a credit with bullet repayment (nominal value 300,000 EUR, 9% interest rate, 3 years duration), setting-up the machine by qualified personnel of the supplier (12,000 EUR) and briefing of the operating personnel (2,000 EUR) in Interval 1, maintenance during the operation stage ('maintenance gold', 10,000 EUR in interval 2 and 3), and guaranteed trade-in at the planned end of operation in interval 3 (-200,000 EUR). The planned lifecycle of the value package comprises 3 years in total.

From these financial data which can be derived from the lifecycle information and outcome attributes of the value bundle model, the tool is able to calculate the original payment sequence of the value bundle selected by ACME Corp. Furthermore, the majority of financial parameters of the investment can be derived from the selected service 'Quick and Easy Credit'. Only information, which differ from customer to customer, such as available equity capital and the interest rate of the open account credit have to be specified by ACME Corp. during the process of calculating the value bundle.

On the basis on the original payment sequence and these financial parameters, the derivative payments for the planned investment can be calculated. To achieve this, all payments in each interval are calculated against the background of the financial parameters, like equity capital, fixed credits and open account credits. Additionally, debit interests and financial investments are calculated.

Finally, the imputed interests applying to the equity capital that has to be expended to finance the investment have to be computed in a separate financial plan. Based on the compiled data, the TCO for the selected value bundle can now be determined (cf. bottom of Figure 6). In addition to the payment sequence, interest rates, imputed interest, and the equity capital to be expended are taken into account. This calculation can be carried out respectively for alternative configurations of the value bundle. Consecutively, the TCO values to be derived from these analyses can be compared to finally select the most adequate value bundle from an economic point of view.

5 Conclusion and outlook

In this paper, we showed that a modelling language for value bundles must account for several requirements. These requirements were systematized in four quadrants (cf. Figure 1), subject to functional and economic issues, as well as subject to the supplier's and customer's point of view.

First, the structure of the value bundles to be offered has to be specified by the supplier. Therefore, the supplier defines the modules to be offered along with configuration rules and a lifecycle model (cf. quadrant (I)). Future work will focus on the evaluation of the proposed language in real-life scenarios, e.g., in terms of usability, effectiveness, and efficiency.

This solution space defined by the supplier is used to limit the possible configurations to be made by customers using the web-based configurator (cf. quadrant (II)). The feasibility of this approach has been shown by the implementation of the proposed modelling language into a meta-modelling tool. Future work will focus on gathering large numbers of model instances which will form the basis for further analysis (e.g. the already mentioned cluster analysis and generation of association rules).

From an economic perspective, the computation of the total cost of ownership of specific value bundle instances from a customer's point of view has been demonstrated (cf. quadrant (IV)). This can be regarded as a contribution to the idea of decision supporting modelling techniques as proposed by vom Brocke [Vomb06]. Future work in this area will focus on the question whether and to what extend suppliers are willing to offer the required information.

The calculation of economic consequences of delivering value bundles (i.e. costs related to the acquisition and application of operand and operant resources—cf. quadrant (III)) is only partially addressed by the current release of our tool. Compared to the other quadrants, this issue requires the most exhaustive extensions to be made.

Acknowledgments

This paper was written in the context of the research projects *FlexNet* and *ServPay*. Both projects are funded by the German Federal Ministry of Education and Research (BMBF), promotion signs 01FD0629 and 02PG1010. We thank the project management agencies *German Aerospace Center* (PT-DLR) and *Forschungszentrum Karlsruhe* (PTKA) for their support. Also, we extend special thanks to the H2-ServPay project seminar students who made the implementation of the presented software tool possible.

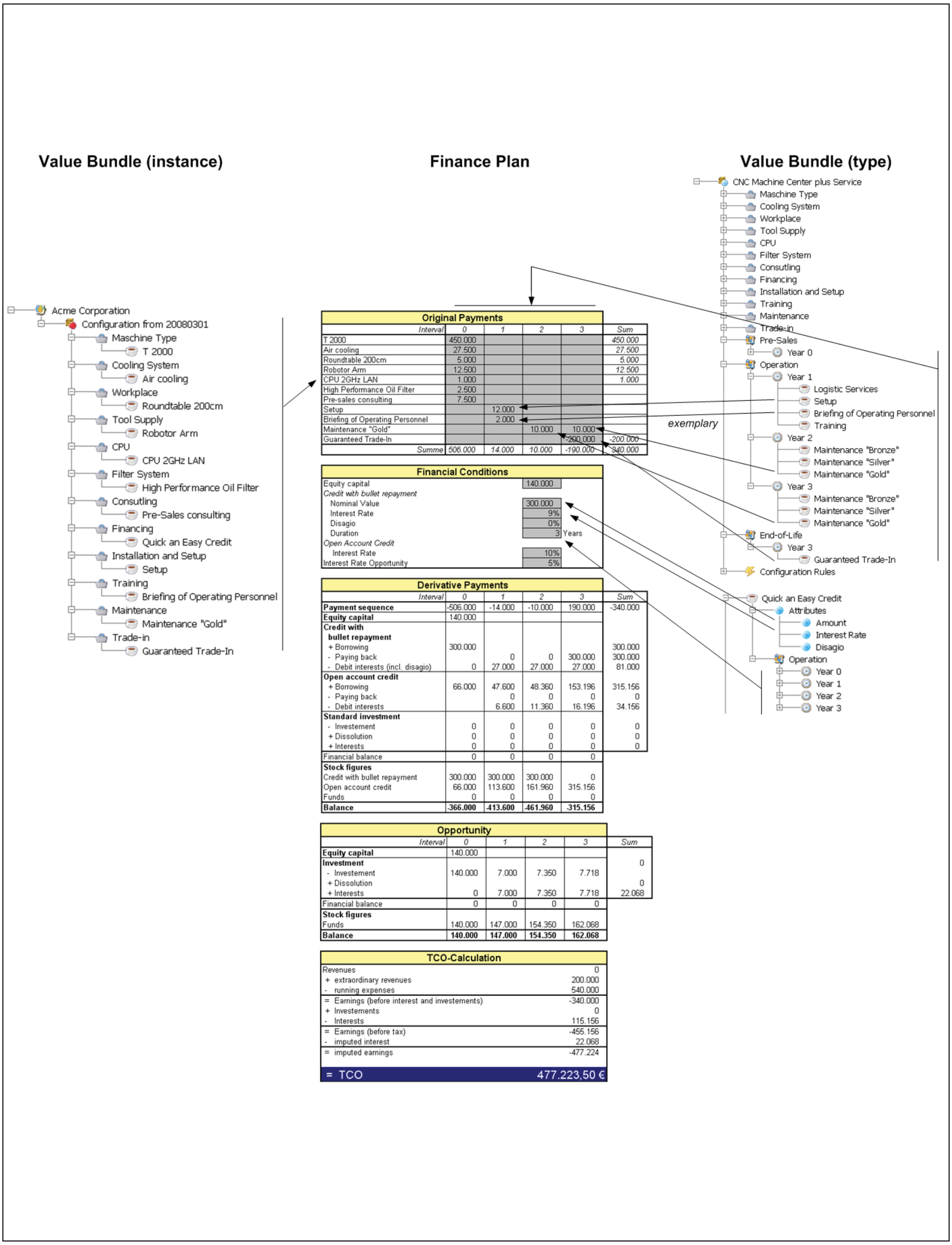

Figure 6: Calculation of the economic consequences (here: total cost of ownership) of a value bundle instance

References

[Alex70] Alexander, C.: Notes on the synthesis of form, 5th Edition, Harvard University Press, Cambridge 1970.

[AnTr00] Anderl, R. and Trippner, D.: STEP: standard for the exchange of product model data, Gabler, Wiesbaden 2000.

[ArGo00] Ardissono, L. and Goy, A.: Tailoring the Interaction with Users in Web Stores, User Modeling and User-Adapted Interaction, Vol. 10 (2000), No. 4, pp. 251-303.

[Arnd78] Arndt, J.: How broad should the marketing concept be?, Journal of Marketing, Vol. 42 (1978), No. January, pp. 101-103.

[BaCl97] Baldwin, C.Y. and Clark, K.B.: Managing in an Age of Modularity, *Harvard Business Review*, September-October 1997, pp. 84–93.

[BaRo08] Basole, R.C. and Rouse, W.B.: Complexity of service value networks: Conceptualization and empirical investigation, IBM Systems Journal, Vol. 47 (2008), No. 1, pp. 53–70.

[BeBe+08] Becker, J., Beverungen, D. and Knackstedt, R.: Reference Models and Modeling Languages for Product-Service Systems – Status-Quo and Perspectives for Further Research, *Proceedings of the 41st Hawaii International Conference on System Sciences (HICSS-41)*, Waikoloa 2008.

[Boeh04] Böhmann, T.: Modularisierung von IT-Dienstleistungen: Eine Methode für das Service Engineering. DUV, Wiesbaden 2004.

[Bott07] Botta, C.: Rahmenkonzept zur Entwicklung von Value Systems, Josef Eul Verlag, Lohmar et al 2007.

[CoGö03] Corsten, H. and Gössinger, R.: Rahmenkonzept zur integrativen Modellierung von Dienstleistungen. In: Schriften zum Produktionsmanagement, No. 58, 2003.

[Emmr05] Emmrich, A.: Ein Beitrag zur systematischen Entwicklung produktorientierter Dienstleistungen. PhD Thesis, University of Paderborn, 2005.

[FiFi01] Fitzsimmons, J.A. and Fitzsimmons, M.J.: Service Management – Operations, Strategy, and Information Technology, 3rd Edition, McGraw-Hill, Boston et al 2001.

[Galb02] Galbraith, J.: Organizing to deliver solutions. In: *Organizational Dynamics*, 31 (2002) 2, pp. 194–207.

[Ganz06] Ganz, W.: Germany: Service Engineering, *Communications of the ACM*, 49 (2006) 7, p. 79.

[GaKu03] Garud, R. and Kumaraswamy, A.: Technological and organizational designs for realizing economies of substitution, In: Garud, R. Kumaraswamy, A., and Langlois, N. (Eds.) Managing in the Modular Age, Architectures, Networks, and Organizations, Blackwell Publishing, Malden et al. 2003, pp. 45–77.

[GrLa04] Grob, H.L. and Lahme, N.: Total Cost of Ownership-Analyse mit vollständigen Finanzplänen. In: *Controlling*, 16 (2004) 3, pp. 157–164.

[Grob89] Grob, H.L.: Investitionsrechnung mit vollständigen Finanzplänen, Vahlen, Munich 1989.

[HaKo07] Hamilton, R.W. and Koukova, N.T.: Choosing options for products: the effects of mixed bundling on consumers' inferences and choices, *Journal of the Academy of Marketing Science*, forthcoming.

[Hilk89] Hilke, W.: Dienstleistungs-Marketing, Gabler, Wiesbaden 1989.

[Holt00] Holten, R.: Entwicklung einer Modellierungstechnik für Data-Warehouse-Fachkonzepte. In: Schmidt, H. (Ed.) Modellierung betrieblicher Informationssysteme: *Proceedings der MobIS-Fachtagung 2000*, Siegen 2000, pp. 3-21.

[Klei07] Klein, R.: Modellgestütztes Service System Engineering – Theorie und Technik einer systemischen Entwicklung von Dienstleistungen, Deutscher Universitäts-Verlag, Wiesbaden 2007.

[KuLo+05] Kunau, G., Loser, K.-U. and Herrmann, T.: Im Spannungsfeld zwischen formalen und informalen Aspekten: Modellierung von Dienstleistungsprozessen mit SeeMe. In: Herrmann, T., Kleinbeck, U. and Krcmar, H. (Eds.) Konzepte für das Service Engineering. Physica, Heidelberg 2005, pp. 149–166.

[Lucz91] Luczak, H.: Service Engineering. Der systematische Weg von der Idee zum Leistungsangebot, TCW Report, Munich 1991.

[Merc03] Mercer: Mercer-Analyse „Service im Maschinenbau" – Ungenutzte Chancen im Servicegeschäft, 2003.

[Mici06] Micic, P.: Das Zukunftsradar. Die wichtigsten Trends, Technologien und Themen für die Zukunft, 2nd Edition, Gabal, Offenbach 2006.

[More02] Morelli, N.: Designing Product/Service Systems: A Methodological Exploration. In: *Design Issues*, 18 (2002) 3, pp. 3–17.

[OECD05] OECD (2005) Enhancing the Performance of the Service Sector, 2005, URL: http://www.value-chains.org/dyn/bds/docs/497/WolflOECDEnhancingPerformanceServicesSector.pdf. Accessed: 2008–02–20.

[Oliv86] Oliver, T.: The real Coke, The real Story, Random House, New York 1986.

[OMG07] Object Management Group (OMG): Production Rule Representation (PRR), Beta 1. http://www.omg.org/spec/PRR/1.0/. Accessed: 2008-08-01.

[Osul06] O'Sullivan, L.: Towards a Precise Understanding of Service Properties, PhD thesis, Queensland University of Technology, Brisbane 2006.

[PaSt+08] Payne, A., Storbacka, K. and Frow, P.: Managing the co-creation of value, Journal of the Academy of Marketing Science, 36 (2008) 1, pp. 83–96.

[Pros07] ProSTEP iViP (2007) Architektur und Aufbau. http://www.prostep.org/de/standards/was/ausbau. Accessed: 2007–12–31.

[Sche06] Scheer, C.: Kundenorientierter Produktkonfigurator: Erweiterung des Produktkonfiguratorkonzeptes zur Vermeidung kundeninitiierter Prozessabbrüche bei Präferenzlosigkeit und Sonderwünschen in der Produktspezifikation, Logos, Berlin 2006.

[ScGr+06] Scheer, A.W., Grieble, O. and Klein, R.: Modellbasiertes Dienstleistungsmanagement. In: Bullinger, H.J. and Scheer, A.W. (Eds.) Service Engineering –

Entwicklung und Gestaltung innovativer Dienstleistungen, 2nd Edition, Springer, Berlin 2006, pp. 19–52.

[Schm08] Schmitz, G.: Der wahrgenommene Wert hybrider Produkte: Konzeptionelle Grundlagen und Komponenten. In: Proceedings of the Multikonferenz Wirtschaftsinformatik, Munich 2008, pp. 665–683.

[Schu98] Schuette, R.: Grundsätze ordnungsgemäßer Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle, Gabler, Wiesbaden 1998.

[Shos77] Shostack, G.L.: Breaking free from Product Marketing. *Journal of Marketing*, 41 (1977) 2, pp. 73–80.

[Shos82] Shostack, G.L.: How to design a service. *European Journal of Marketing*, 16 (1982) 1, pp. 49–63.

[Stil03] Stille, F.: Product-related Services – Still Growing in Importance, *DIW Economic Bulletin* 6, 2003, pp. 195-200.

[StBa+07] Sturm, F., Bading, A. and Schubert, M.: Investitionsgüterhersteller auf dem Weg zum Lösungsanbieter – Eine empirische Studie, Institut für Arbeitswissenschaft und Technologiemanagement, Stuttgart 2007.

[Tebo06] Teboul, J.: Service is Front Stage: Positioning Services for Value Advantage, Palgrave Macmillan, Basingstone et al 2006.

[VaLu04] Vargo, S.L. and Lusch, R.F.: The Four Service Marketing Myths. Remnants of a Goods-Based, Manufacturing Model, *Journal of Service Research*, 6 (2004) 4, pp. 324–335.

[VaLu07] Vargo, S.L. and Lusch, R.F.: Services-dominant logic: continuing the evolution, Journal of the Academy of Marketing Science, Vol. 36 (2007), No. 1, pp. 1-10.

[Vomb06] Vom Brocke, J.: Serviceorientiertes Prozesscontrolling. Gestaltung von Organisations- und Informationssystemen bei Serviceorientierten Architekturen, Professorial Dissertation, University of Münster 2006.

[WiLu06] Winkelmann, K. and Luczak, H.: Modeling, simulation and prospective analysis of cooperative provision of industrial services using coloured Petri Nets, *International Journal of Simulation*, 7 (2006) 7, pp. 10–26.

Jörg Becker, Daniel Beverungen,
Ralf Knackstedt, Oliver Müller

European Research Center for Information Systems,
University of Münster
Leonardo-Campus 3
48149 Münster
Germany
{becker|daniel.beverungen|ralf.knackstedt|oliver.mueller}
@ercis.uni-muenster.de