Four-Worlds Model for Configurable Services

Heiskala Mikko, Tiihonen Juha, Anderson Andreas, Soininen Timo

Abstract

Configurable products are one way to pursue mass customisation. *Configurators* are information systems that support the repetitive specification of product variants and description of all possible product variants of a configurable product in a *configuration model*, Configuration models are expressed using concepts and their relationships defined in the *conceptual model* underlying the configurator. Many different conceptual models exist and there is no dominant approach. The conceptual model significantly affects ease of modelling, understandability, compactness and expressiveness of configuration models as well as computational complexity of computer support. In this paper we present a conceptual model, or conceptualisation, for modelling *configurable services*, an area with relatively little previous research, based on a synthesis of previous work extended with our own experiences with four case companies.

Keywords

conceptual model; configurable services; configurable products; configurator

Contact information

Mikko Heiskala, M.Sc. (Tech), Researcher. Laboratory of Work Psychology and Leadership, Department of Industrial Engineering and Management, Helsinki University of Technology, P.O. Box 5500, 02015 HUT, Finland. Mikko.Heiskala@tkk.fi

Juha Tiihonen, Lic.Sc. (Tech), Researcher, project manager, Software Business and Engineering Institute, Department of Computer Science and Engineering, Helsinki University of Technology, P.O. Box 9210, 02015 HUT, Finland, Juha.Tiihonen@tkk.fi

Andreas Anderson, M.Sc. (Tech), Software Engineer, Variantum Oy, Tekniikantie 14, 02150 Espoo, Finland, andreas.anderson@variantum.com

Timo Soininen, Dr. Sc. (Tech), Docent, Software Business and Engineering Institute, Department of Computer Science and Engineering, Helsinki University of Technology, P.O. Box 9210, 02015 HUT, Finland, timo.soininen@iki.fi

1 Introduction

Today, customers are demanding products that meet their increasingly diverse needs better. Mass customisation has been proposed (Pine, 1993) as a solution to this challenge. *Mass customisation* is the ability to provide products adapted to individual customer needs on a large scale at, or close to, mass production efficiency, using flexible processes (Pine, 1993; da Silveira et al., 2001). One way to pursue a mass customisation strategy is through configurable products. The design of a *configurable product* specifies a set of pre-designed elements and rules on how these can be combined into valid product individuals (Tiihonen & Soininen, 1997). Such knowledge is called *configuration knowledge*. The design of a configurable product is used repetitively, in a routine manner without creative design, in the sales-delivery process to produce specifications of product variants that meet the requirements of particular customers, which is called a *configuration task*. A product configurator, *configurator* for short, is an information system that supports the management and modelling of configuration knowledge and the configuration task (Tiihonen & Soininen, 1997; Sabin & Weigel, 1998).

During the past decades several configurator systems have been developed. These have been employed mostly for mechanical and electronic products, i.e. goods (e.g. Faltings & Freuder, 1998; Soininen & Stumptner, 2003), recently in software as well (e.g. Hotz & Krebs, 2003; Myllärniemi et al., 2005). The focus of attention for such systems has been on suitable product knowledge modelling concepts and formal languages based on these, and correct and efficient inference algorithms for supporting the configuration tasks (e.g. Faltings & Freuder, 1998; Soininen et al., 1998; Felfernig et al., 2001; Soininen & Stumptner, 2003). However, research on *configurable or mass customisable services*, and development of configurators particularly suitable for these, is relatively limited (da Silveira et al., 2001; Heiskala et al., 2006; Harvey et al., 1997; Papathanassiou, 2004; Paloheimo et al. 2004, p. 41, 62; Akkermans et al., 2004; Wimmer et al., 2003; Peters & Saidin, 2000; Winter, 2001; Meier et al., 2002; Dausch & Hsu, 2003; Böhmann et al., 2003). For the purposes of this paper, we define *configurable services* as services that can be customised to individual specifications from a set of options designed to meet a predetermined range of customer needs.

Service has been defined as a process carried out by the supplier as a solution to customer problems, often with customer participation. In literature, services are often argued to have the following characteristics: intangibility, perishability, simultaneity of production and consumption, and heterogeneity – all characteristics goods rarely have (Grönroos, 2000, p. 46-7). Simply adopting product knowledge modelling concepts from goods may not be optimal, because of the aforementioned differences (Wimmer et al., 2003). Moreover, how good a fit a conceptual model has with its intended domain of use affects its quality (Lindland et al., 1994).

In this paper we provide one step towards tool support by presenting a conceptual model for modelling the knowledge important for configuring services in a configurator, from the points of view of the customer and sales person. Our synthesis, the Four-Worlds Model, both extends and simplifies previous work to achieve a conceptualisation that we hope to be practical for modelling and implementable in tool support. We build on ideas of a synthesis of the few existing formally defined proposals for modelling configurable services (Akkermans et al., 2004; Wimmer et al., 2003), service literature (see Section 3) and previous configuration conceptualisations. The synthesis is supported by our experiences on how such services could be modelled in four case companies that we have been working with for the past two years. The companies represent machinery maintenance, insurance and financial services, and telecommunications services.

The rest of the paper is structured as follows. In the next section we present our research goals and methodology, our synthesis of the previous work and our case experiences follows. In section 4 we present the Four-Worlds Model and in section 5 we describe our experiences in modelling case company services. Discussion and comparison with previous work is followed by the conclusions and directions for future work that end our paper.

2 Research goals and methodology

Our research problem was the following: *What kind of a conceptual model can be used to model configurable services from a customer and sales perspective?*

To answer the problem we set out to define a conceptualisation suitable for such a purpose either by developing a new one or by refining, extending, or synthesising existing conceptual models. We sought to define a conceptual model that would *(1)* support eliciting customer needs, by customer themselves or sales personnel or the two in cooperation, in a sales situation and possibly over the web and *(2)* fit the service domain i.e. describe configurable services with concepts that are expressive enough to capture relevant knowledge and are understandable to domain experts.

We reviewed the extant literature for configuration conceptualisations, service modelling and services in general in order to identify the relevant concepts for modelling configurable services. We have used both conceptual analysis of the found existing models and literature and our experiences in our case companies, to extend and synthesise previous work in our conceptualisation, Four-Worlds Model for configurable services. In addition to the insurance service example we use in section 4 we evaluated the Four-Worlds Model by modelling the configurable services of two cases representing equipment maintenance agreements and broadband subscriptions.

3 Synthesis of previous work and case experiences

Here we discuss the literature and models related to services and configurable services in particular. We present the synthesis of the previous work and our case experiences that serves as the basis for the four modelling perspectives of our model, see section 4. For a summary of the previous work, see Table 1.

Services are often defined as processes. E.g. in the Grönroos's (2000, p. 46) definition, service is a process taking place usually between the customer and the different resources of the service supplier. Further, in service design literature it is argued that the service concept should, among other things, describe how the company intends to satisfy customers' needs and with what kind of resources (Edvarsson & Olsson, 1996; Grönroos, 2000, p.193; Goldstein et al., 2002). Processes and their necessary resources are described in service design models of Bullinger et al. (2003), Scheer et al. (2004) and Ma et al. (2002), in the service configuration model of Akkermans et al. (2004) and also in Dausch & Hsu's (2003) reference model for mass customisation of service agreements of heavy equipment maintenance. A distinguishing feature of service delivery processes is customer participation, suggesting that the participation has to be managed to achieve good service process quality (e.g. Grönroos, 2000). Hence, we argue that a conceptual model for configurable service should incorporate means to describe the service process.

Table 1 Previous work and its synthesis as the four worlds

Reference: *description*	**Objects-of-service world**	**Needs world**	**Service solutions world**	**Process world**
Grönroos, 2000, p. 46, 51, 63: *service marketing and management book*	"Service is a process.. ..that .takes place in interactions between the *customer* and service employees.. ..provided as solutions to *customer* problems."	"Service is a process.. ..provided as solutions to *customer problems.*"	"Service is a process.. ..provided as *solutions* to customer problems." Service management deals with process (how received) and *outcome (what received)* consumption.	"Service is a *process..* ..that normally.. ..takes place in *interactions between the customer* and.. *..resources..* ..of the service provider.." Service management deals with *process (how received)..* ..consumption.
Edvarsson & Olsson, 1996; Grönroos, 2000, p.193; Goldstein et al., 2002: *service design and the service concept*		The service concept describes what *customer needs* the service company intends to satisfy, i.e. the *benefits* for the customer,…	…*what* is to be done for the customer…	… *how* this is to be achieved and *with what* kind of *resources.*
Bullinger et al., 2003; Scheer et al., 2004: *service design models*		Service outputs may be aligned to the *customer needs* or target groups (only Scheer et al., 2004).	Describes *what the services include* i.e. the *outcomes or outputs* of a service.	Describes the *process, how* the service outcomes are achieved, necessary *resources,* and *customer integration* in process modules.
Ma et al., 2002: *a service design model, focus on customer process and experience*	A *customer* can be an individual, group, or an organisation and *can be characterised with attributes.*		The customer experience of the *outcomes of process activities* form the customer perceived *service benefit.*	Model describes the service *process* especially from the viewpoint of *how the customer participates* in it.
Akkermans et al., 2004: *a formally defined service configuration conceptual model*		*Value* perspective expresses the *customer needs and demands.*	*Offering* perspective describes what a supplier offers to its customers.	*Process* perspective describes *how* the service is put into operation, includes *resources and customer participation* in the process.
Wimmer et al., 2003: *a formally defined model for financial services mass customisation*	Argue that in financial services relevant *customer attributes should be modelled.*		Financial services are based on contractual *agreements,* which their model describe.	
Winter, 2001: *a simple financial services configuration conceptual model (not formally defined)*	Stresses the importance of including *customer properties* in the configuration.		*Product variants* described with product types, their attributes and attribute values.	
Dausch & Hsu, 2003: *a reference model for mass customisation of equipment maintenance service agreements*	*Characteristics of the maintained equipment* and its *environment* may affect the service agreement.	Model includes a strategic tier describing the *goals of* both *the customer* and provider.	Goals are linked to corresponding parts of the *service agreement,* which resides on the tactical tier.	Tactical tier also *describes the processes* required to fulfil the service agreement and *the necessary resources.*
Soininen et al. 1998: *product configuration conceptualisation*		Functions describe what the product can be used for or the *needs it satisfies.*	*Product characteristics* are described by several concepts.	
Felfernig et al. 2000; 2001: *product configuration conceptualisation*		Functions describe the functionality of the product i.e. its *use purposes*	*Product characteristics* are described by several concepts.	

Services are carried out for the service recipient (Grönroos, 2000). The recipient can be e.g. a person, organisation, a technical system, or a piece of equipment as in (Dausch & Hsu, 2003) and in our maintenance service cases. Wimmer et al. (2004) and Wimmer (2001) argue that modelling customer properties is important for configuration of financial services. Our findings in the insurance case support their views. In fact, our insurance and financial case uses an IT tool in some sales situations to collect relevant customer characteristics and subsequently recommend suitable insurance policies and financial instruments. Further, the installation environment of the maintained machinery affects both the service process and available service options in the maintenance case. In the broadband subscription case, properties of the customer's PC and the geographical location of customer's residence affect the availability of service options. Therefore, we argue that modelling characteristics of the customer, service recipient(s), and their relevant environment should be possible. Such information can be useful for the service process, can affect the service agreement options, and can be used to identify and recommend a suitable service and service options for the customer.

In service design (e.g. Edvarsson & Olsson, 1996; Grönroos, 2000, p.193; Goldstein et al., 2002; Scheer et al., 2004) the needs the service is supposed to satisfy are often defined and spelt out. Similar issues are modelled in (Akkermans et al., 2004) with their value perspective and in product configuration conceptualisations of Soininen et al. (1998) and Felfernig et al. (2000; 2001) with function concepts. Dausch & Hsu (2003) model the customer goals the service is supposed to meet. The aforementioned IT tool of our financial service case also elicits customer needs that are then used to recommend suitable insurance policies. The website of the broadband subscription case uses information about the intended use of the subscription to recommend a suitable service. Hence, we argue that modelling customer needs is useful in sales configuration of services and should be possible.

A description of what the customer receives, the service product or outcomes, is present in all of the studied models and literature. In our case companies, the descriptions are service agreements, contracts. We therefore argue that modelling the service specification should be possible.

4 Four-Worlds Model for Configurable Services

Here we present the Four-Worlds Model for Configurable Services on a conceptual level with a configuration modelling focus. We use the Unified Modeling Language (UML) (e.g. Booch et al., 1999) to define the model semi-formally. We have defined a service configuration modelling language (SCML) based on the Four-Worlds Model (Anderson, 2005), and a mapping from SCML to a modelling language of research prototype configurator WeCoTin (Tiihonen et al., 2003) that is based on the conceptualisation of Soininen et al. (1998). This allows us to configure services in WeCoTin. The concept of using the four worlds to describe configurable services on a more general level (e.g. for design purposes), SCML, and the mapping to WeCoTin will be described in later publications.

4.1 Overview of the Four-Worlds Model

The Four-Worlds Model aims to capture configurable services at sufficient and appropriate detail for the sales stage. The model is divided to four viewpoints we call *worlds*, see Figure 1, each having their own main concepts and dependencies between them. The *objects-of-service world* describes the recipient(s) of service (like persons or physical systems) and the environment relevant to the recipient(s) and would typically include the customer. The objects-of-service world

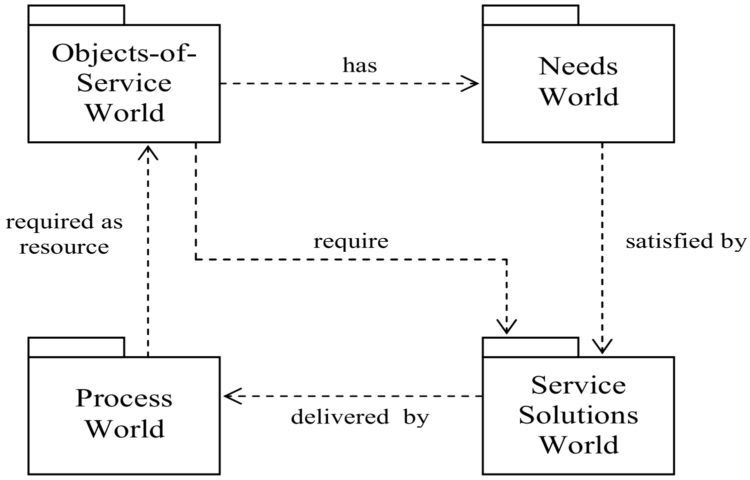

Figure 1 Overview of the Four-Worlds Model

aims to specify *what the company needs to know* about the service recipient(s) and the environment to be able to successfully configure the service and later on deliver the service accordingly. The *needs world* captures the reasons *why* a customer would want to buy the service, e.g. the benefits sought from the service and the needs it satisfies. Together with the objects-of-service world it can be helpful in identifying a suitable solution for a particular customer. The *service solutions world* describes the specifications to which the service is to be delivered. It describes *what* the customer can buy and will eventually be delivered. The service solutions world is the core of the model. The *process world* depicts the service delivery process and resources required to carry it out at appropriate detail for the sales stage. It is intended for communicating the process for the customer (possibly participating in it), not for detailing operation procedures for the company employees. The process world describes *how and with what* the service is put into practice. Communicating the process to customers and especially their role in it could help to better manage customers' participation in the process and keep customers' expectations realistic, both possible contributors to service quality and customer satisfaction (Grönroos 2000, p. 221).

There can be dependencies between the worlds. A service recipient with given characteristics usually has certain needs and requires a specific service solution. The needs are satisfied with particular solutions. The solutions are delivered by given processes and involve certain resources. The service recipient may be a more or less active participant in the process, thus being required as a resource in the process.

Example: We demonstrate the Four-Worlds Model using a simplified example loosely based on insurance services of one of our case companies, Tapiola Group, a prominent group of Finnish mutual insurance and financial services companies. Tapiola's use of a tool helping insurance clerks to identify relevant customer characteristics and needs and subsequently recommend suitable insurances based on these has in part motivated our objects-of-service and needs worlds.

The objects-of-service world entails a customer and a car. The age of the car must be known as it affects the possible insurance coverage and may affect needs.

The needs world describes motoring-oriented needs of the customer. In case of an accident or a breakdown, it is possible to specify the desired level of assistance and whether collision damages should be covered.

The service solutions world consists of a car insurance solution that includes mandatory car insurance, and optionally one of two types of voluntary car insurance. Both include coverage against theft, fire, etc., and collision damage coverage is optional. Budget voluntary car insurance is available only for cars that are at least six years old. It is less expensive, because cheapest accepted (third-party) parts are used for repairs and the insurance company decides where and when the car will be repaired. Extended voluntary car insurance is available only for cars that are 10 years old at the most. It has optional road assistance with two levels, both levels cover towing costs and the insurance company arranges towing. Extended road assistance adds compensation for expenses to continue the trip.

The process world defines a process in case of a car breakdown or an accident. If towing is needed, the insurance company arranges it if the car has extended voluntary car insurance. Otherwise the customer must arrange the towing himself. After the towing the car must be repaired. The repair process depends on the type of the voluntary car insurance. The budget repair process is applied for cars with budget voluntary car insurance, and the normal repair process is applied to cars with extended voluntary car insurance.

4.2 General modelling concepts

In this section we introduce the general modelling mechanisms. A *configuration model* defines a configurable service and the possibilities of customising it with *types* and their *properties*. We distinguish between *TYPES* and their instances occurring in a *configuration*, i.e. *INDIVIDUALS*. A configuration describes a specific service instance. A configuration is said to be *correct* with respect to the product's configuration model, if it is *complete* and *valid*. A configuration is complete if all necessary selections are made, and valid if no rules are violated (Peltonen et al. 1998).

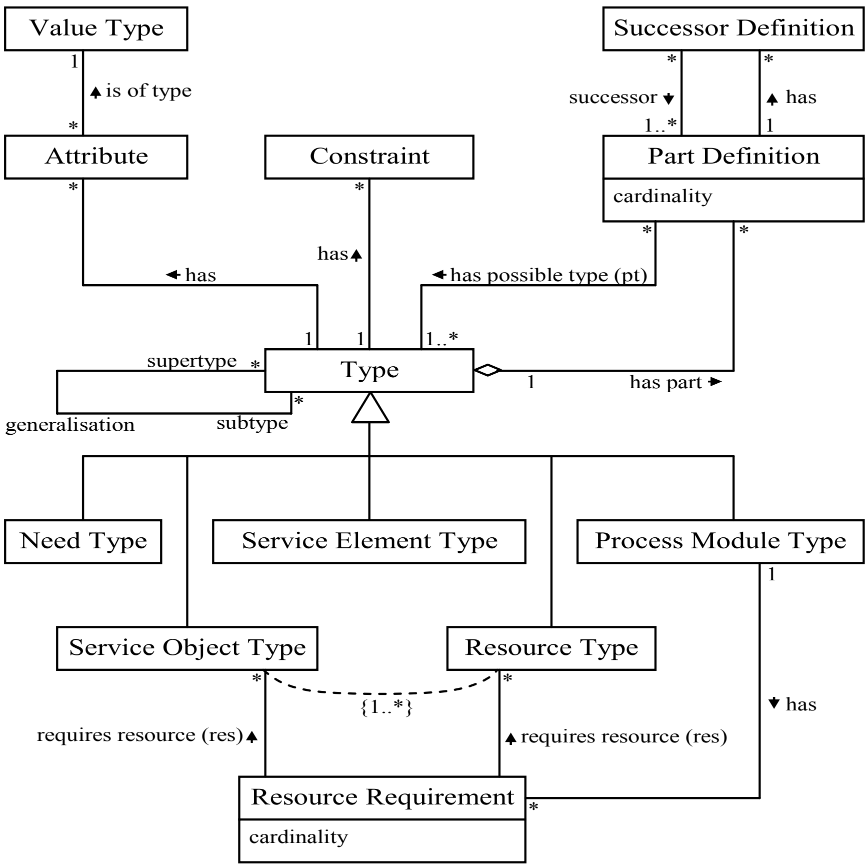

Figure 2 Metamodel

For each world, at most one type can be designated as the *configuration type* of the world. An individual of a configuration type acts as the root of the compositional hierarchy in a configuration for that world. At least one configuration type must be defined in a configuration model.

We use UML in the metamodel (Figure 2) to define stereotypes corresponding to the modelling concepts. Configuration models contain classes that are instances of these stereotypes (Figure 3 (a)). A configuration contains instances of classes of a configuration model (Figure 3 (b)).

Types have a unique *name*. Each type is either *abstract* or *concrete*. Only individuals of concrete types can appear in a configuration. Types can be organised in *generalisation* hierarchies. Types in a generalisation hierarchy must be of the same direct subtype of *Type*, the root type of the generalisation hierarchy, e.g. all supertypes of a need type must be need types as well. A *subtype* inherits the properties, i.e. *parts*, *attributes,* and *constraints*, in addition to *resource requirements* (see 4.6) in case of process module types, of its *supertypes*. Subtypes are said to be *direct subtypes* of the supertypes directly above them in a generalisation hierarchy. Those subtypes that are not direct subtypes are *indirect subtypes*. *Direct* and *indirect supertypes* are defined analogously. Multiple inheritance among types is allowed. All subtypes of a concrete type must be concrete. A generalisation hierarchy cannot contain cycles. No supertype can be both a direct and indirect supertype of type. A subtype may *refine* the properties it inherits from its supertypes in a restrictive manner. As a result each type is replaceable by all its subtypes.

ATTRIBUTES characterise types. A type defines for its attributes an *attribute name,* the *possible values* the attribute can have through its *VALUE TYPE,* a *fixed* or *default value,* and a *necessity definition*. There are four *basic value types*: *integer* (integer numbers), *float* (floating-point numbers), *string* (character sequences), and *Boolean* (true or false). The necessity definition indicates whether the attribute is *optional* or *obligatory*. An optional attribute does not require a value in a complete configuration whereas an obligatory attribute does. For an attribute with a fixed value, all individuals of the type have the specified fixed value. If a default value is assigned to an attribute, the specified value is used in a configuration if an individual of the type does not assign another value to the attribute. Default values are either *proper* or *tentative*. A tentative default value requires a confirmation in a configuration whereas a proper default does not.

Types describe their compositional structure with parts. Types in a compositional structure must be of the same direct subtype of *Type*. Part are specified with *PART DEFINITIONS* that define a *part name*, a *set of possible types* that can occur as the part, a *cardinality* describing the number of individuals that must occur as the part, and a *default* or *fixed realisation*. Default and fixed realisations are defined analogously to attribute default or fixed values. If the cardinality allows multiple individuals, their similarity can be specified. The individuals can be ‘any’ combination of the possible types or directly of the ‘same type’, or even ‘identical’. Additionally, parts of process module types define their possible *successors* (see subsection 3.6).

Types define *CONSTRAINTS* that specify conditions that must hold in a consistent configuration. Constraints can be used to model arbitrarily complex interdependencies of types, individuals and their properties when other concepts are not sufficient to capture an aspect of a service. Constraints are either *hard* or *soft*. A hard constraint must always hold whereas a soft constraint can be violated. We assume there is a constraint language with sufficient expressive power.

4.3 Objects-of-service world

The main concept of the objects-of-service world is *service object*. A *SERVICE OBJECT TYPE* is an

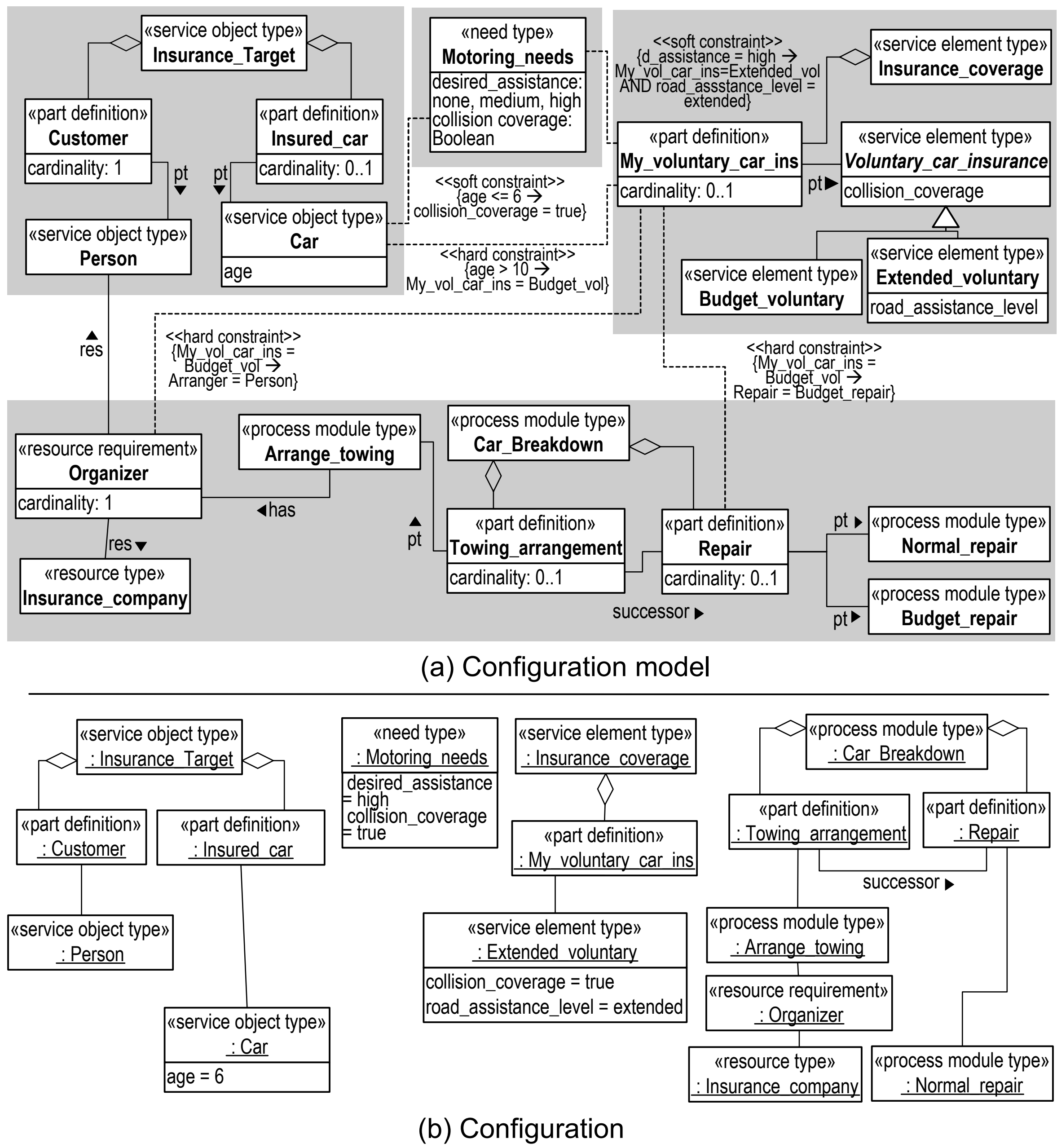

Figure 3 (a) Example configuration model (b) Example configuration

entity representing a service recipient (like persons or physical systems) or environment relevant to the recipient. Examples for the compositional structure of service objects could be a family and its members or maintained equipment and its structure.

Example: Figure 3 (a) shows the configuration model of our running example, and Figure 3 (b) exhibits a configuration. *Insurance_target* represents objects-of-service. It contains exactly one *Person* in the role of a *Customer*. The role of an *Insured_car* specifies properties of the customer's car to be insured. The configuration has a *Person*, and a *Car* with 6 years of age.

4.4 Needs world

Need is the main concept in the needs world. A *NEED TYPE* denotes a benefit sought from a service by a customer. Basis for compositional structure of needs can be e.g. decomposing general needs to more detailed ones, like a general need of being reachable at all times decomposed to being reach-able by phone, fax, or email.

Example: The needs world describes if collision damages coverage is desired, and the level of assistance in case of an accident or a breakdown. This is modelled with attributes of *Motoring_needs*. The example *Motoring_needs* specifies a *high desired_assistance*, and *collision coverage* is included.

4.5 Service solutions world

The service solutions world is centered on a *service element*. A *SERVICE ELEMENT TYPE* describes a part of the pre-delivery service specification, i.e. agreement, about what is to be delivered. Examples of service elements and their compositional structure could be messaging services decomposed to SMS, MMS, fax, and email messaging.

Example: *Insurance_coverage* has optionally a *Voluntary_car_insurance* that can be of type *Budget_voluntary* or *Extended_voluntary*. *Extended_voluntary* can have road assistance in case of a car breakdown or an accident. Road assistance is available only for cars whose age is 10 years at the most. *Budget_voluntary* does not include road assistance, and it is available only for cars that are 6 years or older. Accident repairs are made with non-original parts in a repairs shop decided by the insurance company. In *Extended_voluntary* the customer decides where the car is repaired and original parts are be used. The configuration contains *Insurance_coverage* with *Extended_voluntary* with included *collision_coverage* and *extended road_assistance_level*.

4.6 Process world

The central concept of the process world is *process module*. A *PROCESS MODULE TYPE* represents a task, which could be carried out as part of the service delivery process. A process module may require specific *resources* to be successfully carried out. In its *RESOURCE REQUIREMENT*, a process module defines a set of *possible types, cardinality*, *similarity*, and a *fixed* or *default realisation*. These are analogous to those of parts. Resources may only be service object types or *RESOURCE TYPES*. A resource type describes a physical thing, information, a person, or something else that is necessary for the execution of process modules.

In the process world part definitions take on added semantics: the *precedence* of tasks in a process is defined with part definitions and their *successors*. A successor defines the parts that can follow the defining part in the process. A *SUCCESSOR DEFINITION* includes a *name*, set of *possible successor parts*, *necessity*, and *fixed* or *default successor*. All possible successor parts must be parts of the same defining process module type. The necessity indicates whether a successor is *optional* or *obligatory*. An obligatory successor requires a successor part in a complete configuration whereas an optional does not. If a successor is specified with a fixed successor part, all individuals of the defining type have this part as a successor. Default successors are analogous to default part realisations, e.g. *proper* or *tentative*.

The semantics of the compositional structure of process modules is that the execution of a process module individual means that the process module individuals as its parts are executed as well. For example, machinery repair could decompose to parts Notify of fault, Identify fault, Obtain spare parts (either from customer managed on-site stock, if available, or own supply), and Repair fault. Of these, Notify of fault could require the customer as a resource depending on whether the maintained machinery has remote fault diagnostics installed or not.

Example: Either the insured or the insurance company arranges towing. Thus, *Arrange_towing* requires as *Organizer* resource either *Insurance_company* or a *Person* denoting the insured. Arrangement of towing can be succeeded in the process with *Repair*, which is either *Budget_repair* or *Normal_repair* depending on the chosen voluntary insurance type. The configuration has *Towing_arrangement* with *Insurace_company* as the *Organizer* resource, and *Repair* is managed with *Normal_repair* process. We have modelled successor definitions in Figures 3 (a) and 3 (b) only with a named association due to limitations of space.

4.7 Dependencies between worlds

The dependencies are modelled with constraints. Examples can be found in Figure 3 (a).

Objects-of-service – needs world: Certain service objects, or service objects with given properties, *have* specific needs. For example, a single person has different insurance needs compared to a parent with a family to care for. ***Example:*** *collision_coverage* is recommended for cars that are at most six years old.

Objects-of-service – service solutions world: Further, similarly as above, service objects with given properties *require* certain solutions, i.e. service elements. For example, a customer owning a boat often requires a boat insurance whereas a customer without a boat does not. ***Example:*** The age of the *Insured_car* affects available service elements, e.g. *Extended_voluntary* insurance is available only for cars whose *age* is at most 10 years.

Needs – service solutions world: Needs are *satisfied by* specific service elements. For example, a need of being reachable at all times is satisfied by SMS, MMS, and email access with mobile phone services. ***Example:*** If customer desired high level of assistance *Extended_voluntary* is recommended, this is modelled with a soft constraint.

Service solutions – process world: Specific service elements with given properties are *delivered by* certain process modules. For example, a mobile voice mail service is delivered by the company enabling the service at their end and then by customer taking it in use, e.g. initializing passwords. ***Example:*** If *My_voluntary_car_ins* is of type *Budget_voluntary*, *Repair* uses the *Budget_repair* process.

Objects-of-service – process world: Service objects are *required as resources* for process modules. For example, as above, a customer is required to perform actions to take a mobile voice mail in use, like initializing passwords. ***Example:*** A *Person* denoting the customer is required as *Organizer* resource, if *My_voluntary_car_ins* is of type *Budget_voluntary*.

5 Modelling experiences

In addition to the running example, the Four-Worlds Model has been evaluated by modelling the services of two of our case companies. Here we briefly discuss our modelling experiences.

5.1 Case Broadband Subscriptions

We modelled broadband subscriptions based on public information that a telecommunications company offers on its website. Therefore some crucial aspects of the service may be missing.

Modelling was rather straightforward after the analysis was done using the four worlds. The division to the four worlds seemed useful when modelling. It provided a separation of concerns that was welcomed as the configuration model was quite large. The model contains 59 types, 72 attributes, 20 parts, and 31 constraints. Further, it felt natural to model information contained into a particular world with concepts specifically describing the modelled phenomena. The knowledge acquisition from the website took about two days, planning the model a day and a half, and writing the model half a day. The model was written from a customer self-service viewpoint.

Information modelled to objects-of-service and solutions world represent selections that can be currently configured in the company's online shop. The needs world consists of selections that guide the customers to choosing the most appropriate service. Currently, needs analysis is available as a separate web-based tool, the customers must memorise its recommendations in the online shop. The process world contains information about the delivery process that the customers currently have to interpret according to the selections they have made during configuration. The information is used to communicate the delivery process and their role in it to the customers. Modelling the process world did reveal a shortcoming in the concepts. The process world has four process modules that have a customer contact as a resource requirement. During configuration a new customer contact resource individual would have to be created and defined for each module. This could be avoided by modelling resource requirements similarly to Soininen et al.'s (1998) ports instead of partonomic relationships.

5.2 Case Equipment Maintenance and Repair

The information for the modelling was collected from several interviews with company employees and customers and from the company contract and marketing material.

Modelling the maintenance service was not quite as straightforward as in the broadband case, probably in part due to the newness of the maintenance services to the modeller compared to broadband subscriptions. The division to the separate worlds was considered useful. The configuration model was relatively large as well, even if the process world was not used. The model contains 28 types, 66 attributes, 22 parts, and 34 constraints. The process world was not modelled because the customers' participation in the maintenance process is minimal and the company intentionally keeps the process as standard as possible. We are not able to give an accurate estimate for the total effort used in knowledge acquisition as it spans a time period of nearly two years. The model was written from a consultative selling viewpoint and intended for internal use of the company. This necessitated a quite long time for the authors to familiarise with the company's services, customer needs, and objects-of-service. However, for the solutions world the knowledge acquisition was done in a quite straightforward manner reading the contract material and took about a day. In turn, planning the model took three days and writing it another two.

Information modelled to the objects-of-service world contains information e.g. about the equipment's type, condition, usage and installation environment and parts of it was readily available in company documents, other parts were revealed in the interviews. This information can be used to anticipate customer needs, plan the maintenance process and find a suitable solution. For the needs world, most of the information was collected from the interviews and completed from the company material. The solutions world describes the service contracts of the company and was quite easy to model. The information was readily available in the service contract forms.

6 Discussion and comparison with previous work

The Four-Worlds Model for Configurable Services is still in development. Although the cases discussed in section 5 represent real services, the modelling was performed by the authors, which creates author bias. For more comprehensive evaluation case company employees should be involved in the modelling and real customers during configuration and more diverse services modelled. Although we have defined and implemented a modelling language for the Four-Worlds Model, a modelling tool for it is lacking. Further, a sales configurator designed especially to support process configuration and the recommendation capabilities of objects-of-service and needs worlds should be implemented. Hence, we present the Four-Worlds Model as an idea to be further developed and more rigorously tested. Nevertheless, in our view the Four-Worlds Model provides a preliminary synthesis of previous models, extended and simplified according to our experiences. Based on the initial feedback from our cases it would seem that the Four-Worlds Model allows modelling these services in a uniform way. Further, the model appears to support structured thinking about configurable services in the case companies, which could prove helpful for communication, documentation, and design as well.

Based on our modelling experiences, we argue that the division of the model to the four worlds is beneficial in several ways. In addition to the separation of concerns that was helpful when modelling both cases described in section 5, the worlds allow for some flexibility. If modelling a world is not necessary for the service in question it can be left out, as was done for the maintenance and repair case. However, the solutions world will probably always be present. From another viewpoint, a model with analogous worlds could be used for mechanical products as well. Customer characteristics and needs may influence preferred product options in mechanical products and could perhaps help in eliciting what customers truly require from the product. Capturing the manufacturing process might be relevant for products whose configuration decisions depend on manufacturing constraints like availability of components.

Service configuration, possibly supported with configurators, has been at least a partial goal in several papers. Of these, Winter (2001), Meier et al. (2002), Böhmann et al. (2003), and Jiao et al. (2003) do not define a conceptualisation for services but do discuss service modelling issues. Dausch & Hsu (2003) propose a reference model for mass customising maintenance services. It is not directly intended for configuration and possibly lacks generalisability with a focus on one domain. Akkermans et al. (2004) and Wimmer et al. (2003) present formally defined conceptual models of configurable services and thus resemble our work most closely.

To the best of our knowledge, no previous model of configurable services provides specific concepts to model customer or service recipient characteristics, an important issue judging from our cases and previous work, see section 3 and Table 1. Therefore we see the objects-of-service world as an extension to previous approaches.

Previous work contains similar ideas to our needs world, see Table 1. However, in our view Four-Worlds Model provides more variability modelling concepts for needs than other service-oriented approaches, like constraints to denote dependencies between needs.

Similar issues as our service solutions world are present in all of the previous work, see Table 1. This seems reasonable as it describes the service agreement, which is what is configured at the sales stage judging from our service cases. In (Jiao et al., 2003; Dausch & Hsu, 2003; Meier et al., 2002) the components of agreements seem to represent processes or their outputs. We, however, aim to clearly distinguish between the agreement options and the processes necessary to deliver them. In our service cases the agreement options generally do not directly represent processes or their outputs.

In most of the service oriented previous work the service process and customer participation in it are either modelled or considered important, see Table 1. Wimmer et al. (2003) and Winter (2001) do not have process modelling concepts perhaps due to their focus on financial services. Our aim in modelling processes is to be able to communicate the process for the potential customer at sales stage. This could help to better manage customers' participation in the process and keep customers' expectations of the service realistic, both possible contributors to better service quality (Grönroos, p. 111-2, 2000). More complex modelling than ours may be needed if detailed information on the scheduling and production of the service are required.

We use similar modelling concepts as do Soininen et al. (1998) and Felfernig et al. (2000; 2001). For example, in our model the concepts compositional structure, generalisation, refinements, default values, attributes, and value types follow closely those of Soininen et al. (1998). The process world has different concepts compared to them. However, our process world contains concepts not present in (Soininen et al., 1998; Felfernig et al., 2000; 2001). We are yet to come across a need for resource production and consumption they use for physical products. Akkermans et al. (2004) do not use generalisation, differing from us and (Wimmer et al., 2003; Soininen et al., 1998; Felfernig et al., 2000; 2001). The port-like connections present in (Soininen et al., 1998; Felfernig et al., 2000; 2001) would seem to be useful for modelling services as well and should be added to the Four-Worlds Model in the future.

Most of the 30 vendors studied in (Anderson, 2005) claim their configurators support services. Only two vendors describe their modelling concepts and neither introduces any service specific concepts. The claimed support might stem from the fact that from the customers' point of view the used modelling concepts are irrelevant. That is, services and the selections in them are displayed during sales configuration to customers using service product specific terms and not using the modelling concepts. As for modelling, the used concepts should represent the phenomena being modelled as much as possible (Lindland et al., 1994).

7 Conclusions and future work

We have proposed a conceptual model for modelling the knowledge important for configuring services in a configurator, from the points of view of the customer and sales person, taking a step towards filling a gap in configuration research. Our Four-Worlds Model provides a preliminary synthesis of previous models, extended and simplified according to our experiences. However, the model needs to be more rigorously tested on more diverse services with company employees in modelling and real customers during configuration. Further, tool support needs to be implemented for both modelling and sales configuration. Nevertheless, on the basis of initial feedback from our case companies the Four-Worlds Model seems suitable for configuration modelling of their services. We have kept what constitutes a correct and complete configuration and pricing out of the scope of this paper, both issues in need of future work.

References

Akkermans, H., Baida, Z., Gordijn, J., Peña, N., Altuna, A. & Laresgoiti, I. (2004). Value Webs: Using Ontologies to Bundle Real-World Services. *IEEE Intelligent Systems, 19(4),* 57-66.

Anderson, A. 2005. *Towards Tool-Supported Configuration of Services*. Master's Thesis. Helsinki University of Technology, Department of Computer Science and Engineering. 2005.

Booch, G., Rumbaugh, J. & Jacobson, I. (1999) *The Unified Modeling Language User Guide*.

Addison-Wesley.
Böhmann, T., Junginger, M. & Krcmar, H. (2003). Modular service architectures: a concept and method for engineering IT services. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, 74-83.
Bullinger. H-J., Fähnrich, K-P. & Meiren, T. (2003). Service engineering – methodical development of new service products. *International Journal of Production Economics, 85,* 275-287.
Dausch, M. & Hsu, C. (2003). Mass-Customize Service Agreements for Heavy Industrial Equipment. *IEEE International Conference on Systems, Man and Cybernetics, 5*, 4809-4814.
Edvardsson, B. & Olsson, J. (1996). Key Concepts for New Service Development. *The Service Industries Journal, 16(2)*, 140-164.
Faltings, B. & Freuder, E.C. (1998). Configuration (Guest editor introduction). *IEEE Intelligent Systems & Their Applications, 13(4)*, 32-3.
Felfernig, A., Friedrich, G.E. & Jannach, D. (2000). UML as domain specific language for the construction of knowledge-based configuration systems. *International Journal of Software Engineering and Knowledge Engineering, 10(4),* 449-469.
Felfernig, A., Friedrich, G.E. & Jannach, D. (2001). Conceptual modeling for configuration of mass-customizable products. *Artificial Intelligence in Engineering, 15(2)*, 165-176.
Goldstein, S.M., Johnston, R., Duffy, J.A. & Rao, J. (2002). The service concept: the missing link in service design research. *Journal of Operations Management, 20,* 121-134.
Grönroos, C. (2000). *Service Management and Marketing: A Customer Relationship Management Approach, 2nd Edition*. John Wiley & Sons, Ltd.
Harvey, J., Lefebvre, L.A. & Lefebvre, E. (1997). Flexibility and technology in services: a conceptual model. *International Journal of Operations & Production Management, 17(1)*, 29-45.
Heiskala, M., Paloheimo, K-S. & Tiihonen, J. (2006). Mass Customisation of Services: Benefits and Challenges of Configurable Services. In *Frontiers of e-Business Research 2005, Proceedings of the e-Business Research Forum conference (eBRF 2005)*, Tampere, Finland, 206-221.
Hotz, L. & Krebs, T. (2003). Supporting the Product Derivation Process with a Knowledge-based Approach. In *Proceedings of Software Variability Management (SVM) – ICSE 2003 Workshop*, Portland, Oregon USA, 24-29.
Jiao, J., Ma, Q. & Tseng, M.M. (2003). Towards high value-added products and services: mass customization and beyond, *Technovation, 23(10)*, 809-821.
Lindland, O.I., Sindre, G. & Solvberg, A. 1994. Understanding Quality in Conceptual Modelling. *IEEE Software, 11(2)*, 42-49.
Ma, Q., Tseng, M.M., Yen, B. (2002). A generic model and design representation technique of service products. *Technovation, 22(1),* 15-39.
Meier, H., Schramm, J.J. & Werding, A. (2002). Development of a stage model based configurator to generate more customer-specific services and to support cooperative service networks. In *3rd CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (ICME2002)*, Ischia (Naples), Italy, 2002.
Myllärniemi, V., Asikainen, T., Männistö, T. & Soininen, T. (2005). Kumbang Configurator - A Configuration Tool for Software Product Families. In *Workshop on Configuration, International Conference on Artificial Intelligence (IJCAI 2005)*. Edinburgh, Scotland.

Paloheimo, K-S., Miettinen, I. & Brax, S. (2004). *Customer Oriented Industrial Services.* Espoo, Finland: Report Series – Helsinki University of Technology, BIT Research Centre.

Papathanassiou, E.A. (2004). Mass customisation: management approaches and internet opportunities in the financial sector in the UK. *International Journal of Information Management, 24(5)*, 387-399.

Peltonen, H., Männistö, T., Soininen, T., Tiihonen, J., Martio, A. & Sulonen, R. 1998. Concepts for Modelling Configurable Products. In *Proceedings of European Conference Product Data Technology Days 1998*, Quality Marketing Services, Sandhurst, UK. 189-196.

Peters, L. & Saidin, H. (2000). IT and the mass customization of services: the challenge of implementation. *International Journal of Information Management, 20(2)*, 103-119.

Pine, B.J. II, (1993). *Mass Customization: The New Frontier in Business Competition*. Boston, Massachusetts, USA: Harvard School Business Press.

Sabin, D. & Weigel, R. (1998). Product Configuration Frameworks – A Survey. *IEEE Intelligent Systems & Their Applications, 13(4),* 42-49.

Scheer, A.-W., Grieble, O. & Klein, R. (2004). Model-based Service Engineering. In *Geberl, S.; Weinmann, S.; Wiesner, D. F. (Eds.): Impulse aus der Wirtschaftsinformatik, 5. Liechtensteinisches Wirtschaftsinformatik-Symposium*, (Physika) Heidelberg, 17-33.

da Silveira, G., Borenstein, D. & Fogliatto, F.S. (2001). Mass customization: Literature review and research directions. *International Journal of Production Economics, 72,* 1-13

Soininen, T., Tiihonen, J., Männistö, T. & Sulonen, R. (1998). Towards a General Ontology of Configuration. *AI EDAM, 12(4)*, 357-372

Soininen, T. & Stumptner, M. (2003). Guest Editorial. *AI EDAM, 17(1)*, 1-2.

Tiihonen, J. & Soininen, T. (1997). *Product Configurators – Information System Support for Configurable Products.* Technical Report TKO-B137. Helsinki University of Technology, Laboratory of Information Processing Science. Also in *Richardson, T. (Ed.) 1997. Using Information Technology During the Sales Visit.* Cambridge, UK: Hewson Group.

Tiihonen, J., Soininen, T., Niemelä, I. & Sulonen R. (2003). A Practical Tool for Mass-Customising Configurable Products. In *Proceedings of International Conference on Engineering Design (ICED 03)*, Stockholm, Sweden, paper 1290 on proceedings CD-ROM.

Wimmer, A., Mehlau, J.I. & Klein, T. (2003). Object Oriented Product Meta-Model for the Financial Services Industry. In *Proceedings of the 2nd Interdisciplinary World Congress on Mass Customization and Personalization (MCPC'03),* Münich, Germany.

Winter, R. (2001). Mass Customization and Beyond – Evolution of Customer Centricity in Financial Services. In *Workshop on Information Systems for Mass Customization (ISMC2001),* Dubai.