Product-Service System Configuration in SOA-Based Environment

Nikolay Shilov

SPIIRAS, 39, 14 line,
199178 St.Petersburg, Russia
nick@iias.spb.su

Abstract. Requirements from rapidly developing global markets force companies to develop new paradigms of business, such as product-service systems. Such paradigms, in turn, require new solutions from Information and Communication Technologies (ICT), and modern ICT do make it possible to develop such new approaches. The paper presents an approach based on efficient management of information services in the open information environment oriented to product-service system configuration. The approach is based on the technologies of ontology and context management. The standards of Web-services are used to provide for interoperability between information services. Application of constraints for knowledge representation makes it possible to integrate with existing services.

Keywords: Service-oriented architecture, ontology, context, product-service system.

1 Introduction

Current trends in the worldwide economy require companies to implement new production and marketing paradigms. This determines major trends of knowledge-dominated economy: (i) shift from “capital-intensive business environment” to “intelligence-intensive business environment” – an “e” mindset – and (ii) shift from “product push” strategies to a “consumer pull” management – mass customisation approach [1].

Product-Service Systems (PSS) assumes orientation on combination of products and services (often supporting the products) instead of focusing only on products. This is a relatively new paradigm that fits well, for example, industrial equipment manufacturers, for which the equipment maintenance is a considerable part of the business. Therefore, tight relationships with customers are of high importance in such cases.

A strategy that brings companies and their customers in a closer collaboration is innovation democratisation. This is a relatively new term standing for involvement of customers into the process of designing and creating new products and services. This makes it possible for companies to better meet the needs of their customers. [2]

W. Abramowicz, L. Maciaszek, K. Węcel (Eds.): BIS 2011 Workshops, LNBIP 97, pp. 184–195, 2011.

For companies with wide assortments of products (more than 30 000 – 40 000 products of approx. 700 types, with various configuration possibilities), it is very important to ensure that customers can easily navigate among them to define needed services. One possible solution is to provide a codification system that can produce easily recognizable and at the same time relatively short codes. This is an important task for customer communication management because well defined and understandable product identification is mandatory for successful collaboration with customers and for ensuring a good corporate look for the company [3-5].

New information technologies open new boundaries for researchers. One of new possibilities is remote usage of information services [6]. The service-oriented architecture (SOA) is a step towards information-driven collaboration. This term today is closely related to other terms such as ubiquitous computing, pervasive computing, smart space and similar, which significantly overlap each other [7].

This paper presents an approach to efficient management of information services in the open information environment for PSS configuration. For this purpose the information is actualized in accordance with the current situation. An ontological model is used in the approach to solve the problem of service heterogeneity. This model makes it possible to enable interoperability between heterogeneous information services due to provision of their common semantics [8]. Application of the context model makes it possible to reduce the amount of information to be processed. This model enables management of information relevant for the current situation [9]. The access to the services, information acquisition, transfer, and processing (including integration) are performed via usage of the technology of Web-services.

Fig. 1 represents the generic scheme of the approach. The main idea of the approach is to represent the product's components by sets of services provided by them. This makes it possible to replace the configuration of PSS with that of distributed services. For the purpose of interoperability the services are represented by Web-services using the common notation described by the application ontology (AO). Depending on the problem considered the relevant part of AO is selected forming the abstract context that, in turn, is filled with values from the sources resulting in the operational context. The operational context represents the constraint satisfaction problem that is used during self-configuration of services for problem solving.

Some elements of the presented approach have been implemented in an industrial company that has more than 300 000 customers in 176 countries supported by more than 52 companies worldwide with more than 250 branch offices and authorised agencies in further 36 countries.

The remainder of the paper is organised as follows. Section 2 describes use cases for the PSS-oriented company. The major components of the approach are described in sections 3 (ontological knowledge representation and formalism of object-oriented constraint networks) and 4 (abstract and operational contexts). The procedure of alignment of Web-service descriptions and the application is explained in section 5. Section 6 summarises main research features of the approach.

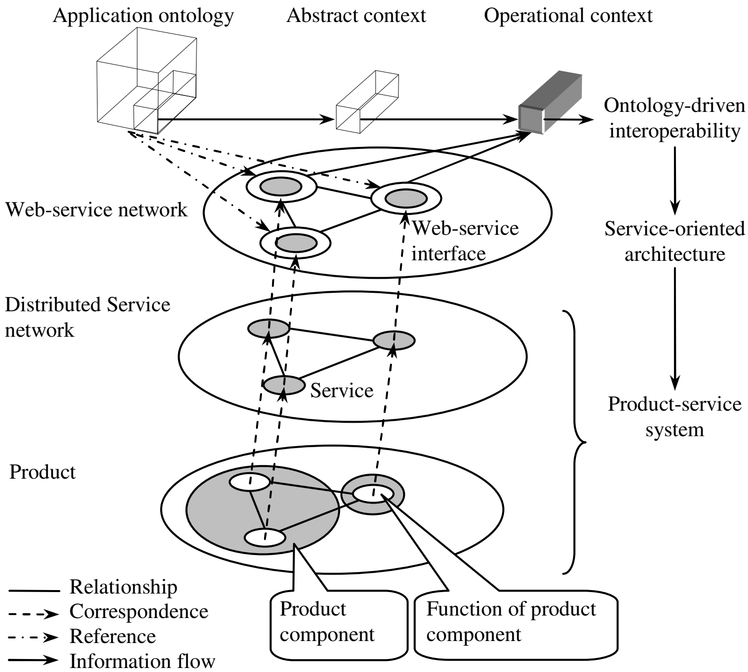

Fig. 1. Generic scheme of the approach

2 Ontology-Driven Use Cases for Industrial Equipment Manufacturer

Ontologies have shown their usability for this type of tasks [e.g., 10-12]. On the basis of the formal description of possible products within the common ontology it is possible now to design new applications which offer customers better ways to find and choose the right products and services.

A simple but always necessary kind of relationship between properties and values describes the consistency of a complex product. This is mainly done by constraints restricting the set of all possible combinations to those which are possible in real-life. The reasons for applying constraints can be different – the most common is the technical possibility of a certain combination.

Furthermore it is possible to add dependant technical data to a certain configuration (which is a set of selected properties and values). For example, a product's weight can be calculated based on the properties / values selected by customer. Another common use case is to configure a CAD 3D model by sending its constructive relevant information from the order code. Practically a lot of data can be made dependant on the current configuration of a modular product. This provides a possibility to provide data which is similarly exact to data of discrete products (for example with a fixed weight).

Even more challenging are inter-product relationships. The most common use case is the relationship between a main product and an accessory product. While both products are derived from a different complex modular product model there are dependencies which assign a correct accessory to a configured main product. Those dependencies are related to the products individual properties and values. The depth of product-accessory relationships is basically not limited, so accessory-of-accessory combinations have to be taken into account, too. Certain problems have to be eliminated like circular relationships which lead back to main product. The relationships can be very complex when it comes to define the actual location / orientation of interfaces and mounting points between products.

A more complex scenario is solution-oriented. The idea is to solve a certain real-life problem with modular-products and their inter-product relationships. The result of such a solution is basically a system of products working together. Used formalism of Object-Oriented Constraint Networks (OOCN, see sec. 3) makes it possible to develop services performing automatic definition of configurable complex products based on the required functions and other constraints specified by the customer.

A handling module offers a good example for this problem. For this purpose the ontology is extended with an extra attribute “Function” for classes.

For illustrative purposes a simple example consisting of three simple products is considered:

- simple product function: movement (Fig. 2a)
- simple product function: rotation (Fig. 2b)
- simple product function: gripping (Fig. 2c)

The goal of the example is to provide for a service to configure complex products that can perform predefined functions. For instance, if two functions (movement and gripping) are required, then the resulting product will consist of two simple products (Fig. 3). If all three functions are required, the resulting product will be as shown in Fig. 4. Such a configuration can be calculated automatically by a constraint solver. Principles of the configuration of more complex constructions are planned to be researched.

The compatibility table for these products is presented in Table 1.

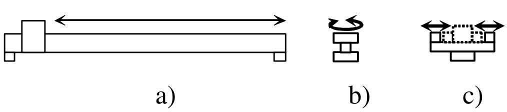

Fig. 2. Simple products: (a) movement function, (b) rotation function, (c) gripping function

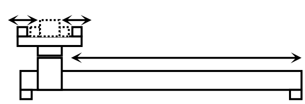

Fig. 3. Complex product performing two functions (movement and gripping)

Table 1. Compatibility table

	Product 1	Product 2	Product 3
Product 1	-	+	+
Product 2	-	-	+
Product 3	-	-	-

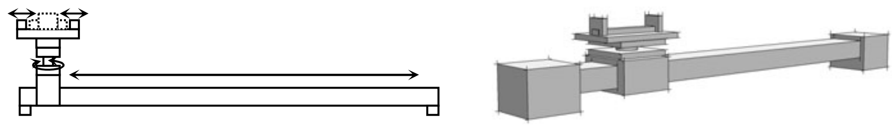

Fig. 4. Complex product performing three functions (movement, rotation and gripping)

3 Ontological Knowledge Representation

In the approach the ontological model is described using the formalism of Object-Oriented Constraint Networks (OOCN). Application of constraint networks allows simplifying the formulation and interpretation of real-world problems which in the areas of management, engineering, manufacturing, etc. are usually presented as constraint satisfaction problems [13]. This formalism supports declarative representation, efficiency of dynamic constraint solving, as well as problem modelling capability, maintainability, reusability, and extensibility of the object-oriented technology.

In the presented methodology the PSS is supposed to be interpreted as a dynamic constraint satisfaction problem (CSP). OOCN provides compatibility of ontology model for knowledge representation and internal solver representations. As a result, ontology-based problem model is described by a set of constraints and can be directly mapped into the constraint solver. A result of CSP solving is one or more satisfactory solutions for the problem modelled.

Compatibility of CSP, ontology, and OOCN models is achieved through identification of correspondences between primitives of these models. CSP model consists of three parts: (i) a set of variables; (ii) a set of possible values for each variable (its domain); and (iii) a set of constraints restricting the values that the variables can simultaneously take. Typical ontology modelling primitives are classes, relations, functions, and axioms. The formalism of OOCN describes knowledge by sets of classes, class attributes, attribute domains, and constraints. Concept "class" in OOCN notation is introduced instead of concept "object" in the way object-oriented languages suggest.

The OOCN paradigm (the detailed description can be found in [14]) defines the common ontology notation used in the system. According to this representation an ontology (A) is defined as: $A = (O, Q, D, C)$ where: O – a set of *object classes* ("*classes*"); each of the entities in a class is considered as an *instance* of the class. Q – a set of class attributes ("*attributes*"). D – a set of attribute domains ("*domains*"). C – a set of *constraints*.

For the chosen notation the following six types of constraints have been defined $C = C^I \cup C^{II} \cup C^{III} \cup C^{IV} \cup C^V \cup C^{VI}$: $C^I = \{c^I\}$, $c^I = (o, q)$, $o \in O$, $q \in Q$ – accessory of attributes to classes; $C^{II} = \{c^{II}\}$, $c^{II} = (o, q, d)$, $o \in O$, $q \in Q$, $d \in D$ – accessory of domains to attributes;

$C^{III} = \{c^{III}\}$, $c^{III} = (\{o\}, True \vee False)$, $|\{o\}| \geq 2$, $o \in O$ – classes compatibility (compatibility structural constraints); $C^{IV} = \{c^{IV}\}$, $c^{IV} = \langle o', o'', type\rangle$, $o' \in O$, $o'' \in O$, $o' \neq o''$ – hierarchical relationships (hierarchical structural constraints) "is a" defining class taxonomy (*type*=0), and "has part"/"part of" defining class hierarchy (*type*=1);$C^V = \{c^V\}$, $c^V = (\{o\})$, $|\{o\}| \geq 2$, $o \in O$ – associative relationships ("one-level" structural constraints); $C^{VI} = \{c^{VI}\}$, $c^{VI} = f(\{o\}, \{o, q\}) = True \vee False$, $|\{o\}| \geq 0$, $|\{q\}| \geq 0$, $o \in O$, $q \in Q$ – functional constraints referring to the names of classes and attributes.

Correspondences between the primitives of ontology model, OOCN, and CSP is shown in Table 2.

Table 2. Correspondence between ontology model, OOCN, and CSP model

Ontology Model	OOCN	CSP
Class	Object	Set of variables
Attribute	Variable	
Attribute domain (range)	Domain	Domain
Axiom / relation	Constraint	Constraint

Below, some example constraints are given:

- the attribute *Locking in end positions* (q_1) belongs to the class *Series C (pneumatic drive)* (o_1): $c^I_1 = (o_1, q_1)$;
- the attribute *Locking in end positions* (q_1) belonging to the class *Series C* (o_1) may take the values *Without (Standard)*, *Extend / Retract*, *Extend*, and *Retract* (the explanation of the values is given in sec. 4): $c^{II}_1 = (o_1, q_1, \{$*Without (Standard)*; *Extend / Retract*; *Extend*; and *Retract*$\})$;
- the class *Valve* (o_2) is compatible with the class *Series C* (o_1): $c^{III}_1 = (\{o_1, o_2\}, True)$;
- an instance of the class *Valve* (o_2) can be a part of an instance of the class *Valve terminal* (o_3): $c^{IV}_1 = \langle o_2, o_3, 1\rangle$;
- the Series C (o_3) is a Pneumatic Drive (o_4): $c^{IV}_1 = \langle o_3, o_4, 0\rangle$;
- an instance of the class *Valve* (o_2) can be connected to an instance of the class *Series C* (o_1): $c^V_1 = (o_2, o_1)$;
- the value of the attribute *cost* (q_2) of an instance of the class *solution* (o_5) depends on the values of the attribute *cost* (q_2) of instances of the class *component* (o_6) connected to that instance of the class *solution* and on the number of such instances: $c^{VI}_1 = f(\{o_6\}, \{(o_5, q_2), (o_6, q_2)\})$.

To summarize, the general three-level scheme of the approach is shown in Fig. 5. The technological base is provided by Web-services. The problem is described via object-oriented constraint networks and the semantics is provided for by using ontologies. Semantic level provides for knowledge sharing and exchange in the PSS model.

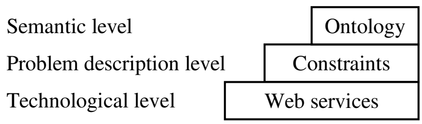

Fig. 5. Three-level scheme of the proposed approach

4 Contexts

For modelling the current state of PSS (current situation) two types of contexts are used: abstract and operational. The abstract context is an ontological model of the current situation build on the basis of selection of knowledge relevant to the current situation. The operational context is a specification of the abstract context for the particular real world situation.

In accordance with the chosen formalism the modelling of the current situation can be defined as the following task. Based on the formalized knowledge about the problem domain $A = (O,Q,D,C)$, build two-level model of the current situation S, presented via the abstract context $Context_a(T_a) = (S|O_a,Q_a,D_a,C_a,R_a,T_a)$ ($O_a \subseteq O$, $Q_a \subseteq Q$, $D_a \subseteq D$, $C_a \subseteq C$, $R_a \subseteq R$, where T_a is the time of the model adequacy) and operational context $Context_p(t) = (S|O_p,Q_p,D_p,C_p,R_p,t)$ ($O_p \subseteq O_a$, $Q_p \subseteq Q_a$, $D_p \subseteq D_a$, $C_p \subseteq C_a$, $R_p \subseteq R_a$, where t is the current time), and find the set of the object instances $I(t) = \{v^o_{q_i} | v \in D_p, o \in O_p, q \in Q_p, i = 1,\dots,N_q\}$, such that all constraints C_p hold, where N_q is the number of attributes of class o, with the parameter values being acquired from the services R of the open information environment at time *t*.

The ontological model of the abstract context is defined as:

$$Context_a(T_a) = (S|O_a,Q_a,D_a,C_a,WS_a,T_a),$$

where S is the time of the situation modelled;

O_a is a set of classes, generally required for modelling the situation S;

Q_a is a set of attributes of the classes O_a;

D_a is a set of domains of the attributes Q_a;

C_a is a set of constraints included into the abstract context;

WS_a is a set of Web-services representing services of the open information environment assigning the values to the attributes Q_a, $WS_a \subseteq WS$, where WS is a set of registered Web-services.

T_a is an estimated time of the model adequacy.

When the information of the open information environment becomes available from the services, references to which are stored in the abstract context, the appropriate values are assigned to the attributes of the classes of the abstract context.

Thus, the operational context is getting built. The operational context $Context_p$ is the model of the current situation in the notation of the object-oriented constraint networks with values assigned to the variables. This model is interpreted as the constraint satisfaction task. The model of the operational context is represented as:

$$Context_p(t) = \left(S \middle| O_p, Q_p, D_p, C_p, WS_p, T_a, \Delta T\right),$$

where t is the current time,

O_p is a set of classes used for the modelling of the situation S in particular conditions,

Q_p, D_p, C_p, WS_p are used sets of attributes, domains, constraints and Web-services respectively;

$\Delta T = t \quad t_0$ is the current time of the operational context life span, where t_0 is the creation time of the abstract context.

The constraint satisfaction task is a triple of sets: $CSP = (V_{CSP}, D_{CSP}, C_{CSP})$,

where V_{CSP} is a set of variables,

D_{CSP} is a set of corresponding domains of the values of the variable,

C_{CSP} is a set of constraints.

The solution of the constraint satisfaction task is a set of values of the variables $v \in V_{CSP}$, such that all the constraints hold. This set contains sets of object instances $I(t) = \left\{ v_{q_i}^{o} \middle| v \in D_p, o \in O_p, q \in Q_p, i = 1, \ldots, N_q \right\}$.

To provide for the information exchange between heterogeneous services as well as its processing, the service model has been developed that is compatible with the ontological model of the knowledge representation. The compatibility is achieved via usage of Web-service interfaces for the services of the open information environment.

5 Alignment of Web-Service Descriptions and the Application Ontology

Illustration of the approach starts with a preliminary phase at that the Web-service descriptions and the AO are aligned. The alignment operation is based on discovering attributes occurred in Web-service descriptions, values of which can assign values to properties of the AO classes. It is supposed that an AO-property can take on a value provided by an attribute the name of which is semantically close to the name of this property. For discovering semantically close names a measure of semantic distance is used.

For the purpose of measuring the semantic distances between concepts containing in the Web-service descriptions and in the AO a machine readable dictionary [15] extracted from Wiktionary [16] is used. Wiktionary was chosen by reasons of its free use, its multilingual support, and keeping, besides lexical relations, definitions of words. The extracted machine-readable dictionary includes 1) a set of words defined in Wiktionary along with for each word 2) definitions given for this word, 3) a set of synonyms, if any, and 4) a set of associated words. Words associated to a word are considered the hyperlinked words occurring in the Wiktionary definition given for this word.

The AO is represented as a semantic network where names of classes and properties specified in the AO constitute nodes of the network. The nodes corresponding to the AO concepts are linked to nodes representing their synonyms and associated words as this is given in the machine-readable dictionary. The links between the nodes are labelled by the weights of relations specified between the concepts represented by these nodes in the machine-readable dictionary. Weight w of a relation specified between two concepts t_i and t_j is assigned as:

$$w = \begin{cases} 0{,}5 & -t_i, t_j \text{ are synonyms} \\ 0{,}3 & -t_i, t_j \text{ are associated words} \\ \infty & -t_i, t_j \text{ are the same word} \end{cases}$$

The values for the weights were evaluated based on the following principles.

Weights for the synonyms are assumed to be greater than weights for the associated words;

Semantic distance is proposed to be calculated as inversely proportional to weights raised to a power. The power corresponds to the path between the compared words. The longer the path the greater the semantic distance for the two different words is expected to be. To meet this expectation with reference to the way of the semantic distance calculation, a weight of the relation between two different words should be in the range (0, 1). Taken into account the first principle the weights 0,5 for the relation between the synonyms and 0,3 for the relation between the associated words are chosen empirically;

The semantic distance between the same words is equal to 0. Correspondingly, ∞ is assigned to the weight of the relation between the same words.

The first step in the alignment operation is parsing a Web-service description represented by the Web Service Definition Language (WSDL). The result of the parsing is a set of meaningful words found in the attribute values of WSDL-tags. If this set contains words differing from the nodes of the semantic network built for the AO, the semantic network is extended with the nodes representing the words extracted from the WSDL-file, synonyms for these words from the machine-readable dictionary, words associated in the machine-readable dictionary with the extracted words, and appropriate links. Only those synonyms and associated words for the WSDL-words are added in the semantic network, which differ from the concept the network already represents. If the semantic network built for the AO contains words found in the WSDL-file, the same words in the AO and in the WSDL file are linked by the relation labelled "∞".

Next, nodes representing WSDL words are checked for their similarity to nodes representing AO-concepts. As a measure of similarity semantic distance *Dist* (2) is used.

$$Dist(t_i, t_j) = \frac{1}{\sum_{S} \prod_{k=s_i}^{s_j} w_k}$$

where t_i - WSDL word, t_j – AO-concept; w – weight of lexical relation existing between t_i and t_j; S – a set of paths from t_i to t_j, where a path s is formed by any number of links that connect t_i and t_j passing through any number of nodes.

After the semantic distances between the words from the WSDL-file and the names specified in the AO are calculated, experts are provided with a ranked list of semantically similar words for each word found in the WSDL-file. Based on this list the experts align related, in their judgment, attribute values in the Web-service descriptions and class properties that can take these values.

Let's consider the following example. The service for finding equipment that meets certain processing requirements has (in its WSDL description) an attribute "*force*" that describes how heavy the processed part can be. In the AO there are two attributes that might correspond to it: "*size*" and "*weight*". An illustrative piece of the semantic network built based on Wiktionary processing and the formula for calculating weights of relationships is represented in Fig. 6.

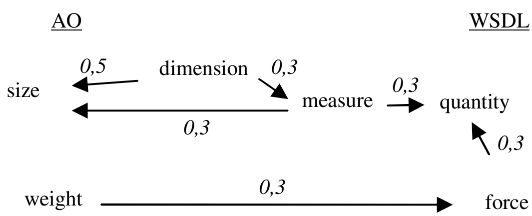

Fig. 6. A piece of the semantic network relevant to WSDL attribute "*force*"

The path from the WSDL word "*force*" to the AO word "*weight"* is "*weight*" → "*force*" (weight 0.3). Thus, semantic distance between the two words is calculated as:

$$Dist(\,force, weight\,) = \frac{1}{0.3} = 3.33\,.$$

The set of paths from the WSDL word "*force*" to the AO word "*size"* in Fig. 6 comprises of two paths and the semantic distance is

$$Dist(\,force, size\,) = \frac{1}{0.3^3 + 0.3^3 \bullet 0{,}5} = 24.69\,.$$

It can be seen that the distance between the concepts "*force*" and "*weight"* is much shorter than between the concepts "*force*" and "*size"*. Hence, the attribute "*weight"* of the AO and the attribute "*force*" contained in the Web-service description can be aligned.

6 Conclusions

The paper presents an approach to integration of efficient management of information services in the open information environment for PSS configuration. The major idea of the approach is product representation via a set of services provided by the product's components. The formalism of object-oriented constraint networks used in the approach makes it possible to represent the problem domain in the approach is described via an ontology with links to the services that provide required information. For each particular situation a fragment of the ontology relevant to the situation is built (the abstract context) and complemented with particular values from the services of the open information environment (the operational context). The operational context, in turn, can be used for problem solving as a constraint satisfaction problem.

The presented codification system significantly simplifies and speeds up the process of product code development in the company producing industrial equipment. Generating a new order code for a new product with the help of the developed system takes approximately one day. The technical options presented by the product manager and developer are converted into order-relevant options. As most of the characteristics can be used again, only new options must be discussed and entered in the system. Without the system this process would need several days. Besides this, the error risk would be very large. It could happen that for the same option another code letter is used for example.

The major advantages of the developed system are:

- Systematic order codes for all products;
- Machine readability;
- Quick orientation for selecting right products and services;
- Security when selecting and ordering products and services.

One other advantage is the reusability of the data. The structured data are used in other processes such as:

- Automatically creating master data in SAP models;
- Automatically creating data for the configuration models and services;
- Automatically generating an ordering sheet for the print documentation (this ordering sheet was generated earlier with high expenditure manually);
- Automatically generating a product and service list which is needed in the complete process implementing new products.

Acknowledgments. The presented work is a result of the joint project between SPIIRAS and Festo Ag&Co KG. Some parts of the research were carried out under projects "COBIT" funded by The Swedish Foundation for International Cooperation in Research and Higher Education, # 09-07-00436 and # 10-07-00368 funded by the Russian Foundation for Basic Research, and project #213 of the research program "Intelligent information technologies, mathematical modelling, system analysis and automation" of the Russian Academy of Sciences.

References

1. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T.: Knowledge Fusion in the Business Information Environment for e-Manufacturing Pursuing Mass Customisation. In: Rautenstrauch, C., Seelmann-Eggebert, R., Turowski, K. (eds.) Moving into Mass Customization. Information Systems and Management Principles, pp. 153–175. Springer, Heidelberg (2002)

2. von Hippel, E.: Democratizing innovation. The MIT Press, Cambridge (2006)
3. Baumeister, H.: Customer relationship management for SMEs. In: Proceedings of the 2nd Annual Conference eBusiness and eWork e2002, Prague, Czech Republic (2002)
4. Fjermestad, J., Romano Jr., N.C.: An Integrative Implementation Framework for Electronic Customer Relationship Management: Revisiting the General Principles of Usability and Resistance. In: Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS 2003), Big Island, HI, USA (2003)
5. Piller, F., Schaller, C.: Individualization Based Collaborative Customer Relationship Management: Motives, Structures, and Modes of Collaboration for Mass Customization and CRM, Working Paper No. 29 of the Dept. of General and Industrial Management, Technische Universität München (2004)
6. Johanesson, P., Andersson, B., Bergoltz, M., Weigand, H.: Enterprise Modelling for Value Based Service Analysis. In: Stirna, J., Persson, A. (eds.) The Practice of Enterprise Modelling, First IFIP WG 8.1 Working Conference, PoEM 2008. LNBIP, vol. 15, pp. 153–167. Springer, Heidelberg (2008)
7. Balandin, S., Moltchanov, D., Koucheryavy, Y. (eds.): ruSMART 2009. LNCS, vol. 5764. Springer, Heidelberg (2009)
8. Uschold, M., Grüninger, M.: Ontologies: Principles, methods and applications. Knowledge Engineering Review 11(2), 93–155 (1996)
9. Dey, A.K.: Understanding and Using Context. Personal and Ubiquitous Computing J. 5(1), 4–7 (2001)
10. Bradfield, D.J., Gao, J.X., Soltan, H.: A Metaknowledge Approach to Facilitate Knowledge Sharing in the Global Product Development Process. Computer-Aided Design & Applications 4(1-4), 519–528 (2007)
11. Chan, E.C.K., Yu, K.M.: A framework of ontology-enabled product knowledge management. International Journal of Product Development 4(3-4), 241–254 (2007)
12. Patil, L., Dutta, D., Sriram, R.: Ontology-based exchange of product data semantics. IEEE Transactions on Automation Science and Engineering 2(3), 213–225 (2005)
13. Baumgaertel, H.: Distributed constraint processing for production logistics. IEEE Intelligent Systems 15(1), 40–48 (2000)
14. Smirnov, A., Levashova, T., Shilov, N.: Semantic-oriented support of interoperability between production information systems. International Journal of Product Development, Inderscience Enterprises Ltd. 4(3/4), 225–240 (2007)
15. Krizhanovsky, A., Lin, F.: Related terms search based on WordNet / Wiktionary and its application in Ontology Matching. In: Digital Libraries: Advanced Methods and Technologies, Digital Collections (RCDL 2009), September 17-21, Petrozavodsk, Russia (2009) (accepted for publication)
16. Wiktionary, the free dictionary (2008), http://en.wiktionary.org/