

Semantic Configuration Web Services in the CAWICOMS Project

Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker

Computer Science and Manufacturing, Universität Klagenfurt Universitätsstrasse 65-67,
9020 Klagenfurt, Austria
{felfernig, friedrich, jannach, zanker}@ifit.uni-klu.ac.at

Abstract. Product configuration is a key technology in today's highly specialized economy. Within the scope of state-of-the-art B2B frameworks and eProcurement solutions, various initiatives take into account the provision of configuration services. However, they all are based on the idea of defining quasi-standards for many-to-many relationships between customers and vendors. When moving towards networked markets, where suppliers dynamically form supply-side consortia, more flexible approaches to B2B integration become necessary. The emerging paradigm of Web services has therefore a huge potential in business application integration. This paper presents an application scenario for configuration Web services, that is currently under development in the research project CAWICOMS¹. An ontology-based approach allows the advertisement of services and a configuration specific protocol defines the operational processes. However, the lack of standards for the semantic annotation of Web services is still a major shortcoming of current Web technology.

1 Introduction

The easy access to vast information resources offered by the World Wide Web (WWW) opens new perspectives for conducting business. State-of-the-art electronic marketplaces enable many-to-many relationships between customers and suppliers, thus replacing inflexible one-to-one relations dating to the pre-internet era of EDI (electronic data interchange). The problem of heterogeneity of product and catalogue descriptions as well as inter-company process definitions is resolved by imposing a common standard on all market participants. The non-existence of a single standard for conducting B2B electronic commerce constitutes a major obstacle towards innovation. Examples for competing and partly incompatible B2B frameworks are OBI, RosettaNet, cXML or BizTalk [25]. They all employ XML² as a flexible data format definition language, that allows to communicate tree structures with a linear syntax; however, content transformation between those catalog and document standards is far from being a trivial task [8]. The issue of marketplace integration mechanisms for *customizable* products is far more complex, because

¹ CAWICOMS is the acronym for "Customer-Adaptive Web Interface for the Configuration of Products and Services with Multiple Suppliers". This work was partly funded by the EC through the IST Programme under contract IST-1999-10688 (<http://www.cawicoms.org>).

² See <http://www.w3c.org/xml> for reference.

products have characterizing attributes that offer a range of different choices. Customers are enabled to configure goods and services according to their individual needs at no extra cost following the paradigm of *mass customization* [23]. Product configuration systems (configurators) support sales engineers and customers in coping with the large number of possible variants and product constellations.

The goal of the research project CAWICOMS is to enable configuration systems to deal simultaneously with configurators of multiple suppliers over the Web. This allows for end-to-end selection, ordering and provisioning of complex products and services supplied by an extended value chain. We employ an ontology-based approach that builds on the flexible integration of these configuration Web services. Furthermore, it can be shown how the capability of each configuration system can be described on the semantic level using an application scenario from the telecommunication domain. For representation of the semantic descriptions the evolving language standard of the 'Semantic Web' initiative [3], [12], OIL resp. DAML+OIL [9] is employed.

In Section 2 we start by giving an overview on the application domain. In Section 3 we describe the Web service architecture and in Section 4 a multi-layer ontology definition for our application domain is given. The interaction processes between the Web service providers and requestors are discussed in Section 5.

2 Application Scenario

Easy access to the corporate network and secure connections to business partners is crucial in today's economy. Virtual Private Networks (VPN) extend the intranet of a possibly multi-national company and are capable of meeting the access requirements at reduced cost using the worldwide IP network services and dedicated service provider IP backbones. VPN infrastructures are designed to be flexible and configurable in order to be able to cope with a rich variety of possible customer requirements. Therefore, the establishment of some concrete VPN involves different steps after determination of customer requirements like locations to be connected or specification of required bandwidth: selection of adequate access facilities from the customer site to some entry point to the VPN backbone, reservation of bandwidth within the backbone, as well as configuration of routing hardware and additional services like installation support.

Note, that it is very unlikely that all these products and services needed for the provision of such a VPN can be supplied by one single organization, but are in general made available by different specialized solution providers, e.g., Internet Service Providers, telecommunication companies or hardware manufacturers (see Figure 1). Therefore, VPNs are typically marketed by specialized resellers (or telecommunication companies like two of our application partners) that integrate the services of individual suppliers and offer complete VPN solutions to their customers.

The integrator/reseller company contracts with the customer and determines - according to the geographic location of the different sites and the qualitative requirements with regards to bandwidth, quality of service or cost limits - the layout of the network service. This configuration task includes the selection of adequate access facilities from the customer site to some entry point of a VPN backbone, reservation of bandwidth within the backbone, as well as parameter setting for routing hardware and configura-

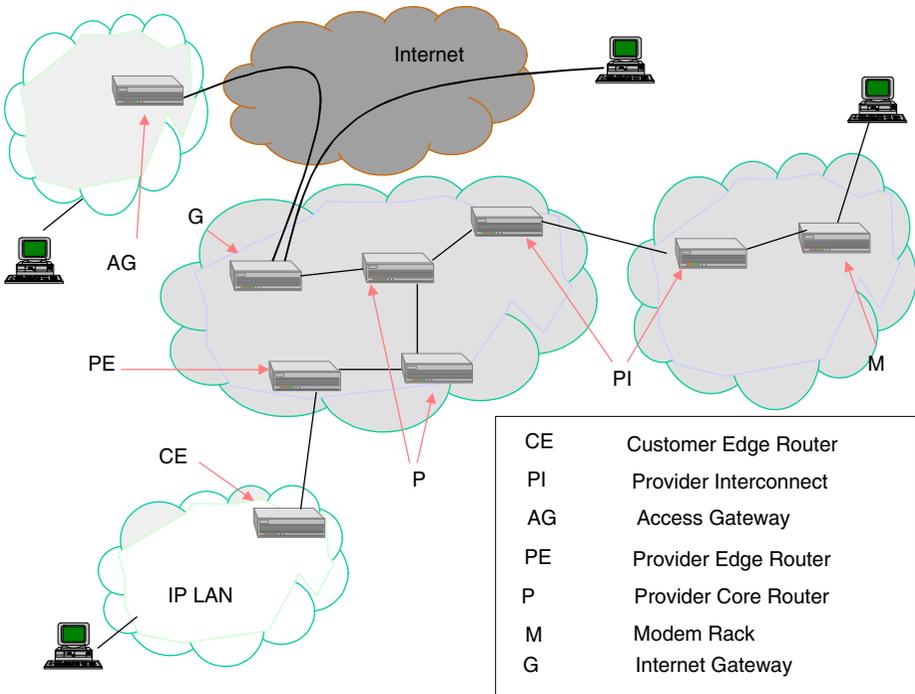


Fig. 1. IP-VPN sketch

tion of additional services like installation support. Considerable parts of this service package will then be sourced from the specialized solution providers [7].

3 CAWICOMS Environment

In the given application scenario, problem solving capabilities are distributed over several business entities that need to cooperate on a customer request for joint service provision. This Peer-to-Peer (P2P) interaction approach among a dynamic set of participants without a clear assignment of *client* and *server* roles asks for applying the paradigm of *Web services* [17]. It stands for encapsulated application logic that is open to accept requests from any peer over the Web.

3.1 Web Services

Basically, a Web Service can be defined as an interface that describes a collection of provided operations. In the following we interpret the application logic that configures a product as a standardized Web service. It can be utilized by interface agents interacting with human users in a Web shop as well as by agents that outsource configuration services as part of their problem solving capabilities. When implementing a Web Service the following issues need to be addressed [17]:

- *Service publishing* - the provider of a service publishes the description of the service to a service registry which in our case are configuration agents with mediating capabilities. Within this registry the basic properties of the offered configuration service have to be defined in such a way that automated identification of this service is possible.
- *Service identification* - the requestor of a service imposes a set of requirements which serve as the basis for identifying a suitable service. In our case, we have to identify those suppliers, that are capable of supplying goods or services that match the specific customer requirements.
- *Service execution* - once a suitable service has been identified the requirements need to be communicated to the service agent that can be correctly interpreted and executed. UDDI, WSDL, and SOAP are the evolving technological standards that allow the invocation of remote application logic based on XML syntax.

Following the vision behind the Semantic Web effort [3,12], the sharing of semantics is crucial to enable the WWW for applications. In order to have agents automatically searching, selecting and executing remote services, representation standards are needed that allow the annotation of meaning of a Web service which can then be interpreted by agents with the help of ontologies.

3.2 Ontologies

In order to define a common language for representing capabilities of configurable products and services we use a hierarchical approach of related ontologies [11,4]. Ontologies are employed to set a semantic framework that enables the semantic description of Web services in the domain of product configuration. Furthermore, we follow the proposal of [10] to structure the ontological commitments into three hierarchy levels (see Figure 2), namely the *generic ontology level*, the *intermediate level* and the *domain level*.

- *Generic ontology level* - Most modeling languages include some kind of meta-model for representing classes and their relationships (e.g. the frame ontology of Ontolingua [11], the UML meta-model [24] or the representation elements of ontology languages such as OIL or DAML+OIL). Such a meta-model can be interpreted as a generic level ontology. Example modeling concepts included in those ontologies are frame, class, relation, association, generalization, etc.
- *Intermediate ontology level* - the basic modeling concepts formulated on the generic ontology level can be refined and used in order to construct an intermediate ontology which includes wide-spread modeling concepts used in the domain. Such an ontology for the configuration domain is discussed in [26] who introduce component types, function types, port types and different kinds of constraints as basic configuration domain specific modeling concepts.
- *Domain ontology level* - finally, using the modeling concepts of the intermediate level, we are able to construct application domain specific ontologies (e.g. network services), which can also be denoted as a configuration models.

Note, that similar approaches to structure ontologies are already implemented in a set of ontology construction environments (e.g. [11]). Our contribution in this context is to illustrate their application for integrating configuration systems.

3.3 Interaction Scenario

In the following we sketch our Web service scenario that focuses on enabling automated procurement processes for customisable items (see Figure 2). Basically there exist two different types of agents, those that only offer configuration services (L) and those that act as suppliers as well as requestors for these services (I). The denotation of agent types derives from viewing the informational supply chain of product configuration as a tree³, where a configuration system constitutes either an *inner node* (I) or a *leaf node* (L). Agents of type I have therefore the mediating functionality incorporated, that allows the offering agents to advertise their configuration services. Matchmaking for service identification is performed by the mediating capability that is internal to each configurator at an inner node. It is done on the semantic level that is eased by multi-layered ontological commitments (as discussed in the preceding subsection) among participants. It is assumed that suppliers share application domain ontologies that allow them to describe the capabilities of their offered products and services on the semantic level. An approach that abstracts from syntactical specifics and proposes a reasoning on the semantic level also exists for transforming standardized catalog representations in [8]. An abstract service description can be interpreted as a kind of standardized functional description of the product⁴. Furthermore, agents in the role of customers (service requestors) can impose requirements on a desired product; these requirements can be matched against the functional product description provided by the suppliers (service providers). If one or more supplier descriptions match with the imposed requirements, the corresponding configuration service providers can be contacted in order to finally check the feasibility of the requirements and generate a customized product/service solution.

4 Multi-layer Ontology Definition

As sketched in Figure 2 the semantic descriptions of the offered configuration services are based on the three layer approach of [10]. The creation of service profiles for each involved configuration system is supported by a set of knowledge acquisition tools, that allow the definition of the product structure with a graphical UML-based notation with precise semantics [5]. Using translators these implementation independent models are translated into proprietary knowledge bases of problem solving engines such as the Java-based *JConfigurator* from ILOG⁵ [14].

However, in the following we will describe our approach employing DAML+OIL as a language for the Semantic Web with precise model theoretic semantics. The correspondence between representation concepts needed for modeling configuration knowledge bases and DAML+OIL is shown in [6]. The uppermost layer of our ontology is the *generic ontology level*. At this level the basic representation concepts and ontological modeling primitives are introduced. These are inherent to the concepts of the modeling language such as *class* and *slot* definitions in OIL. Therefore, it meets the expectations

³ Note, that only the required configuration services are organized in a tree structure, which must not hold for the involved companies in the value chain of a product.

⁴ In [18] this kind of description is denoted as a functional architecture of the configured product.

⁵ See <http://www.ilog.com> for reference.

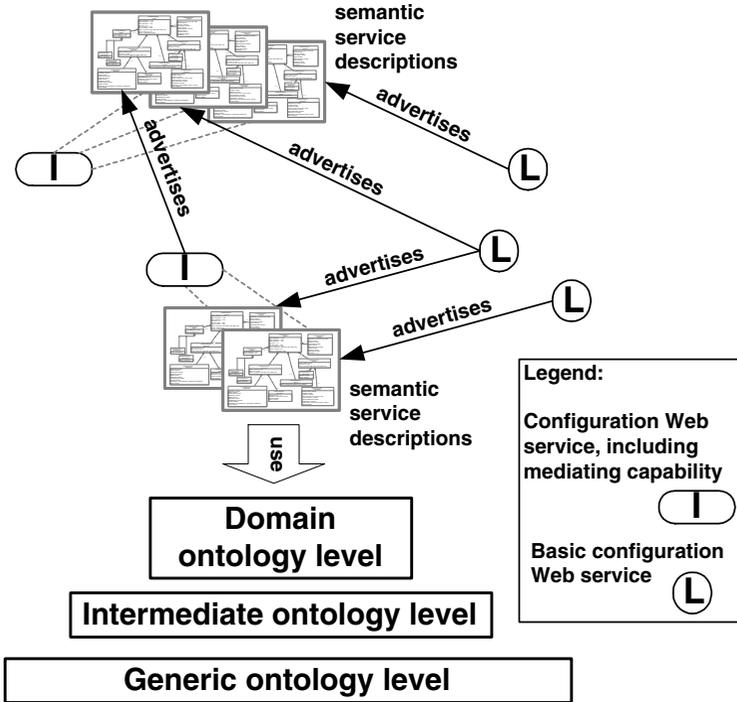


Fig. 2. Web service scenario

towards the uppermost layer and in the following subsection we move on to show which configuration domain specific modeling primitives are to be provided on the *intermediate ontology level*. For reasons of readability *OIL text* [1] is used for representation, i.e. no RDFS-based representation of DAML+OIL is used.

4.1 Basic Configuration Ontology

A general ontology for the configuration domain is important in order to allow easy configuration knowledge reuse and the integration of complex configurable products within marketplace environments. The ontologies proposed by [26] and [5] serve as a basis for the construction of application domain specific ontologies which allow the description of configuration services on a semantic level. Refined concepts of *classes* such as *component types*, *resource types*, *port types*, or *function types* are the basic modeling concepts useful for building the basic product structure. The ontology defined in [26] is based on the frame ontology of Ontolingua [11] and represents a synthesis of resource-based, function-based, connection-based, and structure-based approaches for representing configuration problems. A similar set of concepts is discussed in [5], where the configuration ontology is represented as a UML profile with additional first order formalizations guaranteeing a precise semantics for the provided modeling concepts.

4.2 Product Domain Ontology for Network Services

While the intermediate configuration ontology contains only the basic concepts for modeling product structures, it allows the construction of more specialized ontologies for specific application domains. Furthermore, axioms and slot constraints provided in OIL can be employed to formulate constraints on the configuration model. Exactly these concepts will be refined in the following for representing (application) domain specific ontologies that can be interpreted as a kind of functional product description [18], which is used as a basic framework for formulating capabilities of suppliers and requirements of customers. Figure 3 represents fragments of an ontology for defining configuration services for IP-based Virtual Private Networks (IP-VPN)⁶. For our example we will

```

begin-ontology
ontology -container
title Product Domain Ontology
description IP-based Virtual Private Networks

language "OIL"
ontology-definitions
slot-def protocol
subslot-of
  HasPart //defined in Configuration Ontology
class-def AccessProtocol
subclass-of
  Function //defined in Configuration Ontology

class-def RouterAccess
subclass-of
  AccessProtocol
class-def ModemAccess
subclass-of
  AccessProtocol
class-def InternetAccess
subclass-of
  AccessProtocol
class-def defined Country

class-def defined Town
slot-constraint town_of cardinality 1 Country
class-def LineService
subclass-of
  Function //defined in Configuration Ontology
slot-constraint bandwidth cardinality 1 integer
slot-constraint latency cardinality 1 integer
slot-constraint identifier has-value integer
class-def BackBoneSection
subclass-of
  LineService
slot-constraint access_from has-value AccessLine

class-def defined AccessLine
subclass-of
  LineService
slot-constraint protocol cardinality 1 AccessProtocol
slot-constraint access_to cardinality 1 BackBoneSection
slot-constraint pop cardinality 1 Town
instance-of UK Country

instance-of Manchester Town
related town_of Manchester UK
end-ontology

```

Fig. 3. Domain ontology for IP-VPN services

⁶ The complete example ontology in DAML+OIL can be downloaded from <http://www.cawicoms.org/ontology/ipvpn.rdfs>.

concentrate on the provision of *AccessLines* that connect a customer location (slot *pop* - 'point of presence') to a *BackBoneSection*. The chosen *protocol* (a refinement of the *Has-Part* decomposition relationship in the configuration domain) can be either performed via a router, via a modem or via an internet connection to some access gateway (*Router-Access*, *ModemAccess* and *InternetAccess* are therefore specialized *AccessProtocols*). In addition, an *AccessLine* is characterized by a *bandwidth* and *latency* property that it inherits from its superclass *LineService*, which is in turn a refinement of the *Function* concept (abstract characteristic of a product or service) from the basic configuration ontology (*intermediate ontology level*). The instances contained in the ontology shown in Figure 3 can be interpreted as basic catalog entries representing common knowledge (e.g., British towns or zip codes), which are assigned to base classes of the application domain ontology (in this case *Manchester* is provided as basic instance of the class *Town*).

5 Web Service Scenario

The interaction between service providing agents can be differentiated into the three areas service publishing, identification and execution. As depicted in the scenario in Figure 2, only those agents can request a service that have the mediating capabilities to receive service advertisements and perform service identification.

5.1 Service Publishing

Now we will show how the ontologies defined in Section 4 are used to semantically describe the offered configuration services. Semantic description of the demanded services allows us to implement efficient matchmaking between supply and demand. Within these semantic annotations, restrictions on the domain and cardinality of slots, constraints on connections and structure, as well as the possible types of classes are possible. Furthermore, offered component instances can be represented as subconcepts (e.g. read from a catalog) of the classes of the service domain-specific configuration ontology. Additional supplier-specific constraints are introduced. Consequently, for the semantic description of the capability of a configuration service of a specific supplier the product domain ontology level provides the necessary base concepts that can be further refined. Figure 4 contains the semantic definition of the *AccessLine* services that are offered by the fictitious telecommunication service providers *BTT* and *Luton*. *BTT* serves customers located in the UK and Ireland (constraint on the slot *pop*) and can provide access to *BackBoneSections* 1 through 10 with a maximum *bandwidth* of 2000. In contrast *Luton* offers connections from towns in France and the UK. Only modem or internet are offered protocol choices, a lower *bandwidth* is supported and fewer *BackBoneSections* are accessible. For tailoring the application domain specific configuration ontology to supplier-specific circumstances tool support for acquisition and maintenance of configuration models is needed. Within the CAWICOMS project a *Knowledge Acquisition Workbench* is developed that provides the required tools for designing the service descriptions with a graphical UML-based notation. The generic and the intermediate ontology level as described in Section 4 are inherent to the modeling primitives offered by the tool suite and

```

class-def defined BTT_AccessLine
subclass-of
AccessLine
slot-constraint access_to value-type
  ((slot-constraint identifier value-type (min 1)) and
  (slot-constraint identifier value-type (max 10)))
slot-constraint pop value-type ((slot-constraint town_of value-type (one-of UK)
  or (slot-constraint town_of value-type (one-of Ireland)))
slot-constraint bandwidth value-type (max 2000)

class-def defined Luton_AccessLine
subclass-of
AccessLine
slot-constraint pop value-type ((slot-constraint town_of value-type (one-of France)
  or (slot-constraint town_of value-type (one-of UK)))
slot-constraint access_to value-type
  ((slot-constraint identifier value-type (min 5)) and
  (slot-constraint identifier value-type (max 8)))
slot-constraint bandwidth value-type (max 1200)
slot-constraint protocol has-value (ModemAccess or InternetAccess)

```

Fig. 4. Semantic description of offered services

therefore static in our approach. The tool environment supports human experts in defining and maintaining the application domain specific ontological descriptions as well as in integrating them. The advertisement of the offered configuration services of different suppliers is therefore part of an offline setup process. The functional descriptions of the configurable products and services are communicated to all Web configurators that may act as customers for their configuration service and integrated into their domain ontologies by the human experts.

5.2 Service Identification

Having described service publication, we will now focus on the identification of relevant Web service providers for a concrete demand. This task has similarities with the *surgical* or *parametric search* problem [16], e.g. “*a laptop with at least 20GB hard-disk, 800MHz Pentium III processor or better, manufactured either by Dell or Compaq and costing less than 2000 USD*”. However, for the configuration domain we require even more enhanced search capabilities for identifying the appropriate supply. The reason is, that requirements cannot only be expressed as simple restrictions on product attributes, but also as constraints on the structure. The following example is based on the product domain ontology (Figure 3), requestors are enabled to semantically describe the requested service as can be seen in Figure 5. Let us assume that we search for an *AccessLine* provider that connects us from *Manchester* via *InternetAccess* protocol to *BackBoneSection* ‘3’ with a *bandwidth* of 1200. Here the *bandwidth* slot-constraint is a simple attribute restriction, but the constraint on the slot *access_to* navigates to the related class *BackBoneSection* and restricts the structure. For this example we can intuitively determine that *BTT* is an appropriate supplier for the requested service, as the *Required_AccessLine* qualifies as a subclass to *BTT_AccessLine*. However, for the general case identification of subsumption relationships between offered and required concepts is too restrictive. Consider the case where we would need this *AccessLine* either from *Manchester* or from *Munich*. Assuming all other restrictions remain unchanged, the modified constraint on the slot

```

class-def defined Required_AccessLine
subclass-of
  AccessLine
slot-constraint bandwidth value-type (equal 1200)
slot-constraint pop value-type (one-of Manchester)
slot-constraint protocol value-type InternetAccess
slot-constraint access_to has-value
  (slot-constraint identifier has-value (equal 3))

```

Fig. 5. Semantic description of required service

pop is given in Figure 6⁷. Although *BTT* still provides an appropriate service, the constraint relaxation makes the subsumption of *Required_AccessLine* by *BTT_AccessLine* impossible. So formally the matchmaking task for identification of an appropriate con-

```

class-def defined Required_AccessLine_1
subclass-of
  AccessLine

slot-constraint pop value-type ((one-of Manchester) or (one-of Munich))

```

Fig. 6. Modified service requirement

figuration service can be defined as follows.

Given: A consistent description logic theory T that represents the three ontological layers of our marketplace, a set of concepts $S = \{S_1, \dots, S_n\}$ that describe supply from n different suppliers, and a concept D representing the demanded service.

Task: Identify the set of concepts A , that contains all concepts S_a with $S_a \in A$, where S_a is an appropriate service for D and $A \subseteq S$.

Definition (appropriate service): A service S_a is an appropriate service for D , iff $S_a \cup D$ are consistent.

Note, that this diverges from the approaches taken for matchmaking among heterogeneous agents [27] or for Web service identification [17].

As already mentioned in the previous subsection, the configuration service models are defined within a knowledge acquisition environment and automatically translated into the proprietary knowledge representation formalism of a configuration agent. In our implementation this matchmaking task is therefore performed as part of the search process for a configuration solution of a constraint-based configurator engine. For the internal representation of the advertised service models as well as the service requests an object-oriented framework for constraint variables is employed [14]. Reasoning on service requirements as well as on service decomposition is performed by the underlying Java-based constraint solver. The formulation of service requests and their replies is enabled by a *WebConnector* component that owns an object model layer that accesses the internal constraint representation of the constraint engine. This object model layer represents the advertised configuration service descriptions. A service requestor agent

⁷ Note, that the inherited cardinality constraint restricts slot *pop* to exactly one *Town*, which gives this constraint an *exclusive or* semantics.

imposes its service request via an *edit-query* onto the object-model layer and retrieves the configuration service result via a *publish-query*.

As will be also pointed out in the next subsection, the creation of standards for the definition of semantics of Web services will allow application independent mediating agents to accept service advertisements and to perform the service identification task, which is not the case in the current situation.

5.3 Service Execution

Requests for service execution must conform to an XML-based communication protocol (*WebConnector* protocol) developed for the configuration domain in accordance with the SOAP messaging standard. This protocol defines

- a fixed set of methods with defined semantics for the configuration domain, like creating components, setting values for parameters, initiation of the search process, or retrieving results,
- a mechanism to exchange complex data structures like configuration results and a language for expressing navigation expressions within these data structures (compare to XML-Schema and XPath), and
- extensibility mechanisms for special domains and support for a session concept in HTTP-based transactions.

This way the semantics of the process model of the configuration Web service is defined by a proprietary protocol. This assumption works for our specific requirement of realizing collaborative configuration systems, but is only half way towards the vision of Web services in the Semantic Web. Therefore, markup languages are required that enable a standardized representation of service profiles for advertisement of services as well as definitions of the process model. This way, the task of identifying appropriate services and the decomposition of a service request into several separate requests can be performed by domain independent mediators. Due to the lack of these standards, this mediating functionality is in our case performed by application logic integrated into the configuration systems. DAML-S⁸ is an example for an effort underway that aims at providing such a standardized semantic markup for Web services that builds on top of DAML+OIL.

6 Related Work

Beside standards for representing product catalogs [8], there exists a number of approaches for standardizing electronic commerce communication (e.g. Commerce XML - cXML or Common Business Library - CBL) - these are XML-based communication standards for B2B applications⁹, which also include basic mechanisms for product data interchange and can be interpreted as ontologies supporting standardized communication

⁸ See <http://www.daml.org/services> for reference.

⁹ An overview on existing e-Commerce frameworks for business to business communication can be found in [25].

between e-Business applications. However, these standards are restricted to the representation of standardized products, i.e. the basic properties of complex products, especially configurable products are not considered. Basic mechanisms for product data integration are already supported by a number of state-of-the-art B2B applications. However, the integration of configuration systems into electronic marketplace environments is still an open issue, i.e. not supported by today's systems. Problem Solving Methods (PSMs) [2] support the decomposition of reasoning tasks of knowledge-based systems into sets of subtasks and inference actions that are interconnected by knowledge roles. The goal of the IBROW project [20] is the semiautomatic reuse of available problem solving methods, where a software broker supports the knowledge engineer in configuring a reasoning system by combining different PSMs. A similarity to the work of [20] exists in the sense that the selection of suppliers (and corresponding configuration systems) is a basic configuration task, where configurators must be selected which are capable of cooperatively solving a distributed configuration task. The approach is different in the sense that the major focus is on providing an environment which generally supports a semi-automated reuse of problem solving methods, whereas our approach concentrates on the automated integration of configuration services in an e-business environment. The Infomaster system [15] provides basic mechanisms for integrating heterogeneous information sources in order to provide a unique entry point for the users of the system. Compared to our approach there is no support for the integration of configurable products and the underlying configuration systems. The design of large scale products requires the cooperation of a number of different experts. In the SHADE (Shared Dependency Engineering) project [22] a KIF-based representation [21] was used for representing engineering ontologies. This approach differs from the approach presented in this paper in the sense that the provided ontology is majorly employed as a basis for the communication between the different engaged agents, but is not used as a means for describing the capabilities of agents. The STEP standard [13] takes into account all aspects of a product including geometry and organisational data [19]. The idea of STEP is to provide means for defining application specific concepts for modeling products in a particular application domain. These application specific concepts are standardised into parts of STEP called *Application Protocols* which are defined using the EXPRESS data definition language (Application Protocols are EXPRESS schemas). EXPRESS itself includes a set of modeling concepts useful for representing configurable products, however the language can not be used to define an enterprise specific configuration model without leaving the STEP standard. Similarities to our approach can be seen in the role of application protocols in STEP which are very similar to the domain ontology level discussed in this paper.

7 Conclusions

The Semantic Web [3] is the vision of developing enabling technologies for the Web which supports access to its resources not only to humans but as well to applications often denoted as agent-based systems providing services such as information brokering, information filtering, intelligent search or synthesis of services [20]. This paper describes an application scenario for semantic Web services in the domain of configuring telecom-

munication services. It demonstrates how to apply Semantic Web technologies in order to support the integration of configurable products and services in an environment for distributed problem solving. DAML+OIL-based configuration service descriptions can be used in order to match them with given customer requirements and the matchmaking task to determine the adequacy of a service is defined. DAML+OIL formalisms are well suited for representing the component structure of configurable products, i.e. part-of associations and simple associations between component types and corresponding basic constraints. However, technologies supporting the vision of the Semantic Web are still under development. In order to support a full scenario of distributed configuration Web services, languages like DAML+OIL have to be extended with language elements supporting the formulation of service advertisements as well as process definitions for the interaction.

References

1. S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: A Reason-able Ontology Editor for the Semantic Web. In *Proceedings of Joint Austrian/German Conference on Artificial Intelligence (KI)*, pages 396–408, Vienna, Austria, 2001.
2. R. Benjamins and D. Fensel. . *Special issue on problem-solving methods of the International Journal of Human-Computer Studies*, 49(4), 1998.
3. T. Berners-Lee. *Weaving the Web*. Harper Business, 2000.
4. B. Chandrasekaran, J. Josephson, and R. Benjamins. What Are Ontologies, and Why do we Need Them? *IEEE Intelligent Systems*, 14,1:20–26, 1999.
5. A. Felfernig, G. Friedrich, and D. Jannach. UML as domain specific language for the construction of knowledge-based configuration systems. *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, 10(4):449–469, 2000.
6. A. Felfernig, G. Friedrich, D. Jannach, M. Stumptner, and M. Zanker. A Joint Foundation for Configuration in the Semantic Web. *Technical Report KLU-IFI-02-05*, 2001.
7. A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. Web-based configuration of Virtual Private Networks with Multiple Suppliers. In *Proceedings of the 7th International Conference on Artificial Intelligence in Design (AID)*, Cambridge, UK, 2002.
8. D. Fensel, Y. Ding, B. Omelayenko, E. Schulten, G. Botquin, M. Brown, and A. Flett. Product Data Integration in B2B E-Commerce. *IEEE Intelligent Systems*, 16(4):54–59, 2001.
9. D. Fensel, F. vanHarmelen, I. Horrocks, D. McGuinness, and P.F. Patel-Schneider. OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.
10. A. Gangemi, D. M. Pisanelli, and G. Steve. An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. *Data and Knowledge Engineering*, 31(2):183–220, 1999.
11. T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.
12. J. Hendler. Agents and the Semantic Web. *IEEE Intelligent Systems*, 16(2):30–37, 2001.
13. ISO. ISO Standard 10303-1: Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles. 1994.
14. U. Junker. Preference-programming for Configuration. In *Proceedings of IJCAI, Configuration Workshop*, Seattle, 2001.
15. A. M. Keller and M. R. Genesereth. Multivendor Catalogs: Smart Catalogs and Virtual Catalogs. *The Journal of Electronic Commerce*, 9(3), 1996.
16. D.L. McGuinness. Ontologies and Online Commerce. *IEEE Intelligent Systems*, 16(2):9–10, 2001.

17. Sh. McIlraith, T.C. Son, and H. Zeng. Mobilizing the Semantic Web with DAML-Enabled Web Services. In *Proceedings of the IJCAI 2001 Workshop on E-Business and the Intelligent Web*, pages 29–39, Seattle, WA, 2001.
18. S. Mittal and F. Frayman. Towards a Generic Model of Configuration Tasks. In *Proceedings 11th International Joint Conf. on Artificial Intelligence*, pages 1395–1401, Detroit, MI, 1989.
19. T. Männistö, A. Martio, and R. Sulonen. Modelling generic product structures in STEP. *Computer-Aided Design*, 30,14:1111–1118, 1999.
20. E. Motta, D. Fensel, M. Gaspari, and V.R. Benjamins. Specifications of Knowledge Components for Reuse. In *Proceedings of 11th International Conference on Software Engineering and Knowledge Engineering*, pages 36–43, Kaiserslautern, Germany, 1999.
21. R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12,3:36–56, 1991.
22. G.R. Olsen, M. Cutkosky, J.M. Tenenbaum, and T.R. Gruber. Collaborative Engineering based on Knowledge Sharing Agreements. In *Proceedings of the ACME Database Symposium*, pages 11–14, Minneapolis, MN, USA, 1994.
23. B.J. PineII, B. Victor, and A.C. Boynton. Making Mass Customization Work. *Harvard Business Review*, Sep./Oct. 1993:109–119, 1993.
24. J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1998.
25. S.S.Y. Shim, V.S. Pendyala, M. Sundaram, and J.Z. Gao. E-Commerce Frameworks. *IEEE Computer*, Oct. 2000:40–47, 2000.
26. T. Soininen, J. Tiihonen, T. Männistö, and R. Sulonen. Towards a General Ontology of Configuration. *AI Engineering Design Analysis and Manufacturing Journal, Special Issue: Configuration Design*, 12(4):357–372, 1998.
27. K. Sycara, M. Klusch, and S. Widoff. Dynamic Service Matchmaking among Agents in Open Information Environments. *ACM SIGMOD Record, Special Issue on Semantic Interoperability in Global Information Systems*, 1999.