Unified Service Description Language XG Final Report

W3C Incubator Group Report 27 October 2011

This Version:
http://www.w3.org/2005/Incubator/usdl/XGR-usdl-20111027/
Latest Published Version:
http://www.w3.org/2005/Incubator/usdl/XGR-usdl/

Editors:
Kay Kadner, SAP
Daniel Oberle, SAP
Authors:
Kay Kadner, SAP
Daniel Oberle, SAP
Martin Schaeffler, Siemens
Andrea Horch, IAT University of Stuttgart
Maximilien Kintz, IAT University of Stuttgart
Lukas Barton, HP
Torsten Leidig, SAP
Carlos Pedrinaci, The Open University
John Domingue, The Open University
Massimo Romanelli, DFKI
Ruben Trapero, Universidad Politécnica de Madrid
Konstadinos Kutsikos

This document is also available in these non-normative formats: USDL XG Wiki including larger images.

Abstract

This document is the final report of the USDL XG. The groups goal were:

1. Investigate related work and examine their relation to USDL
2. Reshape the USDL specification in order to be aligned with existing W3C standards and related work
3. Define and implement reference test cases that showcase the use of USDL in practical environments.

The Unified Service Description Language is used for describing business, operational and technical parameters of services. Service descriptions then include information like pricing, legal, service provider, interaction methods, service level agreements and so on. This allows for more sophisticated use cases than service descriptions like WSDL allow today, e.g., comparison of services by price.

The XG's effort was mainly targeted and providing background and validation to strengthen the position of USDL. Therefore, related work from different sources and with different purposes was analyzed and compared to USDL. It turned out that some approaches like SML and SoaML are somehow related to USDL, but follow different purposes and therefore do not fulfill the requirements that USDL fulfills.

Next, we wanted to align USDL better with existing W3C standards. However, as the basis for our work in the XG was a quite complex and comprehensive USDL specification version (USDL3M5), we decided that adoption and validation of USDL is more important than yet another version. We therefore put our focus on the reference test cases. These are test cases that verify USDL from different aspects of use, e.g., Service Provider, Service Host, etc. Not all of the identified roles for test cases where filled. However, the ones that were created come from different companies that shows the public interest in USDL.

The XGR contains the related work analysis, the reference test case specification and implementation description, outlook into future work and recommendations of how to proceed with USDL.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of Final Incubator Group Reports is available. See also the W3C technical reports index at http://www.w3.org/TR/.

This document was published by the W3C Unified Service Description Language Incubator Group as an Incubator Group Report. If you wish to make comments regarding this document, please send them to public-xg-usdl@w3.org. All feedback is welcome.

Publication of this document by W3C as part of the W3C Incubator Activity indicates no endorsement of its content by W3C, nor that W3C has, is, or will be allocating any resources to the issues addressed by it. Participation in Incubator Groups and publication of Incubator Group Reports at the W3C site are benefits of W3C Membership.

Incubator Groups have as a goal to produce work that can be implemented on a Royalty Free basis, as defined in the W3C Patent Policy. Participants in this Incubator Group have made no statements about whether they will offer licenses according to the licensing requirements of the W3C Patent Policy for portions of this Incubator Group Report that are subsequently incorporated in a W3C

Recommendation.

Table of Contents

1 What has been done?

1.1 USDL 3M5 specification

The USDL version 3.0 milestone M5 has been released during the timespan of the Incubator group. Besides the constant smaller improvements that happen in every new milestone, M5 particularly added completely revised versions of the Service Level Module and Legal Module leading to the current modularization depicted in the figure below.

A further change worth mentioning is the introduction of a separate Technical Module which was basically split from the Functional Module. A complete specification for each module as well as an overview documentation was published in March 2011 and is available at www.internet-of-services.com.

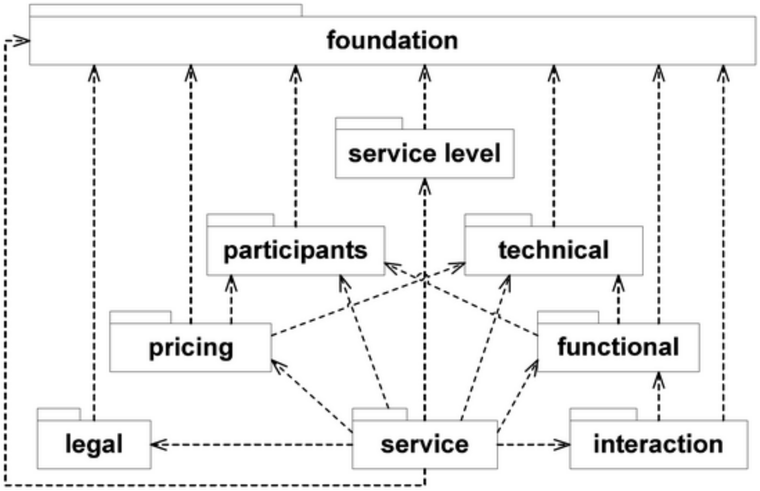


Figure 1: Modules of USDL

A particular outcome of the incubator group has been a US-specific version of the Legal Module for input to future USDL version.

1.2 Review of related work

There exists a plethora of service description efforts that can be grouped into different strands summarized in the table below. Each of the strands has its own motivation and representation needs for capturing service information. The individual efforts can be attributed to the following criteria: (i) whether the scope of the effort lies in capturing IT or business aspects of services or the whole service system. (ii) the purpose of the corresponding effort - is the effort geared towards normative data exchange, is it there to facilitate software engineering, is it there to automate a specific task, or is it there to act as reference model? (iii) whether the effort is able to capture business network relationships between services. (iv) whether the effort is standardized.

Effort	Scope	Purpose	Business network	Representation	Standardized
1. Service Oriented Architectures					
WS-*	IT	Exchange	No	XML	Various
UDDI	IT	Exchange	No	XML	OASIS
SoaML	IT	Engineering	No	UML	OMG
UPMS	IT	Engineering	No	UML	No
SOA-RM	IT	Reference	No	Informal	OASIS
SOA Ontology	IT/ Business	Reference	No	OWL	Open Group
WADL	IT	Exchange	No	XML	W3C
Core Ontology of Web Services	IT	Reference	No	OWL	No
2. Semantic Web Services					
OWL-S	IT	Automation	No	OWL	No
WSMO	IT	Automation	No	WSML	No
...					
SAWSDL	IT	Exchange	No	XML	W3C
SA-REST	IT	Exchange	No	XML	W3C
RO-SOA	IT	Reference	No	RDFS	OASIS
Reference Service Model	IT	Reference	No	OWL	No
WSMO-Lite	IT	Automation	No	RDFS	W3C
MicroWSMO	IT	Exchange, Automation	No	XML	No
Minimal Service Model	IT	Exchange, Automation	No	RDFS	No
3. Software-as-a-Service					
SML	IT	Exchange	No	XML	W3C
SaaS-DL	IT	Engineering	No	XML	No
4. Service Networks					
OBELIX service ontology	Business	Configuration	Yes	RDFS	No
e3Service	Business	Configuration	Yes	RDFS	No
e3Value	Business	Analysis	Yes	RDFS	No
SNN	Business	Optimization	Yes	UML	No
5. Service System					
Alter	System	Reference	No	Informal	No
Reference Architecture Foundation of SOA	System	Reference	No	UML	OASIS
Service Design Model	System	Engineering	No	ECore	No
Ontological Foundations of Service Science	System	Reference	No	FOL	No
TEXO Service Ontology	System/Business	Reference	No	OWL	No
6. Economic					
DIN PAS 1018	Business	Exchange	No	Informal	DIN
Emmrich	Business	Reference	No	UML	No
O'Sullivan	Business	Reference	No	ORM	No
Toma	Business	Automation	No	WSML	No
USDL	Business	Reference / Exchange / Engineering	No	ECore, XML	In progress

SOA Efforts

The first strand of service description efforts is the field of Service-oriented Architectures (SOA). SOA is a way of thinking about IT assets as service components, i.e., functions in a large application are factorized in stand-alone services that can be accessed separately. Because of their IT focus, most approaches limit their attention to the field of software architecture. Originally, several standards bodies specified several dozens of different aspects which are collectively known as WS-* (incl. WSDL, WS-Policy, WS-Security, etc.). Since one of the key components of a SOA is a service registry, the OASIS standards body introduced the concept of Universal Description, Discovery and Integration (UDDI), i.e., a specification for a platform-independent registry. UDDI services shall be discovered via white pages (address, contact, and known identifiers) or yellow pages (industrial categorizations based on standard taxonomies), as well as green pages. However, UDDI does hardly prescribe any schema for such information. As the concept of SOA matured, calls for support in software and service engineering increased. Hence, the OMG standards body dedicated its focus to software engineering for SOA, and, subsequently defined the Service-oriented architecture Modeling Language (SoaML). SoaML actually orginates in the publicly funded EU research projects SHAPE and MODELWARE (Berre 2009). The predecessor which led to the SoaML is the UML Profile and Metamodel for Services (UPMS, (Berre 2008)). Finally, the multitude of description efforts and different definitions of SOA led to a Reference Model for Service Oriented Architecture (SOA-RM) from OASIS (MacKenzie et al. 2006). Similarly, The Open Group drafts an alternative reference model in form of an ontology for Service-Oriented Architectures (SOA Ontology) (Harding 2008).

Current research in the SOA strand mainly concerns RESTful services and their description. Until recently there was no counterpart to WSDL for RESTful services. The W3C submission Web Application Description Language (WADL) is about to fill this gap (Hadley 2009). Note, however, that version 2.0 of WSDL can be used to describe REST Web services, thus competing with WADL. Another recent research is the work of (Oberle et al. 2006) provides an ontological account of Web services. The resulting core ontology of web services is built according to the principles of ontological analysis on top of the DOLCE foundational ontology, and, thus, can also be regarded as a reference model.

Semantic Web Services Efforts

A second strand consists mainly of ontologies in the field of Semantic Web Services. As presented in the seminal paper, viz., (McIlraith et al. 2001), the main goal of Semantic Web Services approaches is automation of discovery, composition, and invocation of services in a SOA by ontology reasoners and planning algorithms. The most prominent efforts are OWL-S (Ankolekar et al. 2001) and WSMO (Roman et al. 2006). Many surrounding and similar efforts have surfaced in literature. For example, (Dobson, Sánchez-Macián) try to unify existing ontologies that capture qualities-of-services in order to automate service discovery. Considering the need to attach semantic descriptions to existing WS-* descriptions and the multitude of ontologies available, the W3C came up with a recommendation called Semantic Annotations for WSDL (SAWSDL) (Farell, Lausen 2007). SAWSDL introduces XML attributes to establish correspondences between tags in WSDL (or arbitrary XML Schema documents) and concepts or relations in an arbitrary ontology. A similar idea to SAWSDL is already adopted by the W3C Member Submission called Semantic Annotations for REST (SA-REST) (Gomadam et al. 2010). SA-REST defines three basic properties that can be used to non-intrusively annotate HTML/XHTML documents, typically to embed ontological meta-data. These properties are included as part of the XHTML document allowing a capable processor to gain extra information about the content of the document.

With the many approaches around came the need to specify a reference model for semantic SOAs. Consequently, the OASIS is also about to specify a Reference Ontology for Semantic Service Oriented Architectures (RO-SOA) (Norton et al. 2008). There also is the recent Reference Service Model (RSM) (Loutas et al. 2011) for the Web of Services that aims to close the gap between two phenomenically contradictory service annotation paradigms: traditional semantic service frameworks and the emerging social annotation of services. RSM aims to (i) facilitate the semantic interlinking between services annotated using different semantic models and (ii) accommodate the bottom-up social annotation of services. RSM was developed following the design science research methodology. To develop RSM, existing semantic service models and SOA service models were reviewed in the light of the six service contracts and examined whether and using which elements each of the models supports in each of the contracts. The identified elements were then fed to a multiphase abstraction exercise. RSM comprises of the following concepts: Service, Service Input, Service Output, Service Context and Service Logic, Service Provider, Service Client and Service Feedback.

Current research topics of this strand tackle the new concepts of Linked Open Data, RESTful services, and Linked Services (Pedrinaci, Domingue 2010). Consequently, the ongoing publicly funded research project SOA4All brought forth WSMO-Lite (Lightweight Semantic Descriptions for Services on the Web) which is in W3C Member Submission state (Fensel 2010). WSMO-Lite consists of only 5 classes to structure the capturing of service semantics in terms of non-functional properties, effects, conditions, and functional classifications. The original purpose of WSMO, namely the automation of discovery and execution, did not change, however. MicroWSMO (Kopecky et al. 2008, Kopecky et al. 2011) is another approach to tackling RESTful services and Web APIs. MicroWSMO is relevant on the syntax level on top of HTML adding "hooks" for pointing to concepts and lifting and lowering definitions. It is similar to SA-REST but relies on a different serialisation (microformat) and acknowledges the need for a service model. The latter is captured in the Minimal Service Model (Pedrinaci, Domingue 2010). This model essentially captures the basic notions of a Web service from a technical point of view. The minimal service model can be used to bring together MicroWSMO, SAWSDL, WSMO-Lite, OWL-S, etc., for instance. It is essentially an integration ontology that covers the basic functional notions of services and is largely a subset of all other service ontologies such as WSMO, OWL-S, etc.

Software-as-a-Service Efforts

The third strand is rooted in the rise of on-demand applications that led to the notion of software-as-a-service (SaaS), covering software applications (e.g., CRM on-demand) and business process outsourcing (e.g., gross-to-payroll processing, insurance claims processing) to cloud and platform services. The emphasis of service here implies that the consumer gets the designated functionality he/she requested together with hosting through a pay-per-use model. Thus, software-as-a-service is not synonymous with SOA. The strand of SaaS contains a standard, namely, the W3C recommendation called SML (Service Modeling Language) (Pandit et al. 2009). SML is a strict superset of XML Schema adding the capability to define constraints on a model (using Schematron rules) and the capability to define and reference model elements in separate files. In addition, SML-IF defines an exchange format so that SML model instances can be easily exchanged between producers and consumers. One anticipated use for SML is to define a consistent way to express how computer networks, applications, servers, and other IT resources are described or modeled so businesses can more easily manage the services that are built on these resources. Therefore, we have classified it in the SaaS strand. Note however, that the use of SML is not limited to SaaS scenarios. Current research is represented by the Software-as-a-Service Description Language (SaaS-DL). SaaS-DL builds on WS-* to capture SaaS specificities in order to support model-driven engineering (Sun et al. 2007).

Service Network Efforts

The fourth strand draws attention mainly to describing Service Networks, i.e., the ecosystem and value chain relationships between services of economic value. So far, this strand did not output any standards and is represented by academic approaches only. An early work is (Baida et al. 2001) which is continued by the ontology of the publicly funded EU research project OBELIX by (Akkermans et al.

2004). The latter is an application ontology that helps users with configuration of service bundling and graphical modeling of service networks. The work continued in (De Kinderen and Gordijn 2008a; De Kinderen and Gordijn 2008b) introduces the e3Service ontology (Akkermans et al. 2004) to model services from the perspective of the user needs. This offers constructs for service marketing, but in a computational way, such that automated reasoning support can be developed to match consumer needs with IT-services. The main focus of this work is to generate service bundles under the consideration of customer needs. Another work by the same group is called e3Value ontology (Gordijn, Akkermans 2001) which focusses on value chains and value exchanges between actors and market segments focussed on business modeling and its analysis. (Bitsaki et al. 2008) introduce the Service Network Notation (SNN) which captures similar aspects to the e^{3}Service ontology. However, SNN is an UML model that can be analyzed for measurements of added value for each single participant as well as for the whole network optimization of value flows.

Service System Efforts

Fifth, there are overarching efforts that concentrate on the bigger picture of service systems or service science also taking into account socio-economic aspects. (Alter 2008) was one of the first to realize that the concept of a service system is not well articulated in the service literature. Therefore, he contributes three informal frameworks as a first attempt to define the fundamentals of service systems. Although the background of the OASIS Reference Architecture Foundation for SOAs (Estefan et al. 2009) is Service-oriented architectures, the specification argues that SOA-based systems are better thought of as ecosystems rather than stand-alone software products. Therefore, the specification is put into the service system category and is directly related to the ontological foundations of service science approach. However, the reference architecture foundation is not based on ontological analysis but takes the OASIS SOA-RM as its starting point by building on its vocabulary of important terms and concepts. Another effort considering the wider scope of the service system is the Service Design Model of (Dhanesha et al. 2009). Other than the aforementioned efforts, the Service Design Model is geared at a software engineering purpose and comes in the form of an ECore model for the Eclipse Modeling Framework (EMF). The model takes into account the business organization, the customer, and the delivery organization during service design. The model aims at providing a foundation for the design for service quality by envisioning different modules (such as financial, resource, process, etc.), configurability, variability, and extensibility. Current research in this strand is represented by the work of (Ferrario and Guarino 2008) which can be seen as a continuation and formalization of Alter’s approach. Although differing in its main notions, they present a reference ontology for ontological foundations of service science which is founded on the basic principles of ontological analysis. In turn, this reference ontology forms the core part of the TEXO Service Ontology which extends it by ontology modules for pricing, legal, innovation, or rating information (Oberle et al. 2009). This ontology is outcome of the German lighthouse research project called THESEUS/TEXO.

Economic Efforts

The final strand is driven by schools of business administration and business informatics and focuses on capturing the purely economic aspects of services regardless of their nature (with less or no focus on IT services and software architectures). Led by the Fraunhofer IAO research institute, the German standard DIN PAS 1018 essentially prescribes a form for the description of services for tendering. The structure is specified in a non-machine-readable way by introducing mandatory and optional, non-functional attributes specified in natural language, such as, classification, resources, location, etc. The standard is driven by needs of the services industry in Germany whose expectations are the professionalization and industrialization of the service industry, the increase of transparency, and eventually the overall development of the service economy. The PhD thesis of Emmrich (2005) has a similar motivation only that this work focuses on product-related services, such as maintenance, and is specified in UML. He basically merges existing standards and models for products, companies, organization, and resources. The PhD thesis of (O’Sullivan 2006) adopts a wider scope and contributes a domain independent taxonomy that is capable of representing the non-functional properties of conventional, electronic and web services. The work compiles the non-functional properties into a series of 80 conceptual models that are categorized according to availability (both temporal and locative), payment, price, discounts, obligations, rights, penalties, trust, security, and quality. The motivation is to provide a reference model for automated service discovery, comparison, selection, and substitution. ORM (Object-Role Modeling) is used to represent the results also available as XML Schema serialization. Toma (Toma 2010) presents a syntactic translation of O’Sullivan’s work in the proprietary WSML language. The goal is to extend the aforementioned WSMO by non-functional properties for automation of discovery, composition, invocation, and, in particular, ranking of services in a SOA (Toma et al. 2008).

USDL

Here, the Unified Service Description Language (USDL) continues by further formalizing and extending the economic approaches. USDL is ongoing research contributed mainly by the TEXO project within the THESEUS research program initiated by the German Federal Ministry of Economy and Technology, projects funded by the German Federal Ministry of Education and Research projects (e.g., Premium Services), as well as EU DG INFSO projects (e.g., FAST, RESERVOIR, MASTER, ServFace, SHAPE, SLA@SOI, SOA4ALL), and the Australian Smart Services CRC.

USDL builds on and acknowledges the standards for the technical IT description efforts for services such as WSDL. However, USDL adds business and operational information on top. In order to achieve this, USDL defines normative UML modules (via the Eclipse Modeling Framework (EMF)) for capturing the "master data" of a service. That includes normative modules, i.e., class models for pricing, legal, functional, participants, interaction and SLA aspects. Therefore, both manual and IT services can be described with USDL. Detailed specifications of each module can be found here.

Unlike any of the other approaches USDL serves a reference purpose, its class models can be used to facilitate model-driven software engineering, and its XSD can be used for information exchange. However, it is mainly the content and normative character of the USDL modules that sets USDL apart from many of the related approaches discussed above. For example, UDDI, WSMO, or OWL-S only prescribe tiny schemata and leave the modeling of service description concepts (such as a generic schema for defining a price model or licenses) to the user. Both W3C SAWSDL and W3C SA-REST are designed to be agnostic of any service description schema. Similar holds for W3C SML.

In order to further show the progression compared to the state-of-the-art, one has to compare USDL on the basis of its individual modules to the plethora of related approaches above. For instance, The pricing module progresses the state-of-the-art compared to (i) the state-of-the-art above and (ii) additional related work of price models independent of the service domain such as (Kelkar et al. 2002) for electronic product catalogs. Therefore, the USDL price model provides a scientific contribution in itself as presented in (Kiemes et al. 2010). Regarding (i), the USDL pricing module allows tiered pricing and a structuring in price components according to established business literature. Both is not possible in (O’Sullivan 2006) and (Toma 2010), for instance. In (De Kinderen and Gordijn 2008a; De Kinderen and Gordijn 2008b), the price model is represented as a mathematical formula included as string in the ontology in order to enable the price determination. Therefore, the work is not directly comparable to ours since the structure of a price plan is not declaratively modeled as discussed in (de Miranda and Baida 2006). Regarding (ii) we refer the reader to (Kiemes et al. 2010), where related approaches are grouped and positioned in four categories (a) economic contributions, (b) established enterprise software, (c) standalone billing engines, and (d) explicitly specified pricing models.

Regarding the Legal Module we surpass (i) (O'Sullivan 2006), and, consequently (Toma 2010) since both are not based on a generic copyright model in a specific legal code. Therefore, they also do not consider geopolitical differences in copyright law. One of the most detailed approaches in dealing with service licensing is from (Gangadharan et al. 2007). They developed a service license model considering traditional software licenses. Based on this model, they have created an ODRL (Open Digital Rights Language) Service Profile (ODRL-S), which can be used to describe service licenses. The service model incorporates the WIPO framework and, therefore, lacks a profound legal foundation. (Speiser 2009) talks about usage policies instead of licenses and identifies different kinds of usage policies. His focus lies on the technical formalization of such usage policies, without any legal foundation. With respect to the body of related work of (ii) we refer the reader to the positioning given in (Baumann and Loes 2010). In general, the domain of services is more diverse than the product domain.

1.3 Specification of use cases

Energy Marketplace

Owner: Martin Schaeffler, Siemens AG

Type of use case: company internal use case

Focus on Role: Service Consumer

Implementation status: planned (waiting for SAP agora)

System Setup

- USDL-Editor M4 (Eclipse PlugIn)
- SAP Service Marketplace (agora)

Roles

- Energy Device
- Energy Service Platform
- Service Provider
- Energy Prosumer
- Service Marketplace

Activities

1. Offer services on the marketplace
    - Service Provider
    - describe and deploy service description
    - USDL Editor and marketplace repository
    - service is descripted and available on the marketplace
2. Subscribe a default “weather” service
    - Energy Service Platform
    - rest service
    - use rest service description to integrate the service
    - Energy Device uses subscribed weather service
3. Log on the marketplace
    - Energy Prosumer
    - with own user account
    - by using webbrower
    - Prosumer is logged on the marketplace
4. Search for services in the catalogue
    - Energy Prosumer
    - filter and browse the catalogue
    - by using the webbrowser
    - find weather services
5. Compare selected services
    - Energy Prosumer
    - use the "compare" functionality in the catalogue
    - by using the browser
    - select the weather service for his/her needs
6. Acknowledge service subscription
    - Energy Prosumer
    - use the "book/acknowledge" functionality in the marketplace
    - by using the browser
    - service booked and booking acknowledged
7. Inform service provider about changed subscriptions
    - Service Marketplace
    - sending booking information of the marketplace customer to the provider of the service
    - by E-Mail or system functionality
    - Service Provider is informed about the booking and has the customer data
8. Change subscription
    - Energy Service Platform

    - by using the deposited data from the marketplace
    - by asking the marketplace about the current subscriptions of the Energy Prosumer via http (?)
    - new subscription is used by the Energy Device
9. See new service in action on the Energy Service Platform
    - Energy Prosumer
    - by logging on the Energy Service Platform
    - by using the webbrowser
    - verify the service in use

Creating and uploading a service

The goal of this test case is the definition, creation and implementation of a weather service, i.e., a service that provides weather forecast for a given time and location. After the service has been implemented, described and finished, it is uploaded to a marketplace and made publicly available.

Owner: Kay Kadner, SAP

Type of use case: basic, enabling technology

Focus on Role: Service Engineer

Implementation status: done

System Setup

- USDL Editor
- USDL Marketplace
- Service Development Environment (e.g., Eclipse)

Roles

Service Engineer

Activities

The service engineer (SE) of a company has the task to provide a service via USDL.

1. Create the business logic
    - The Service Engineer
    - uses
    - Eclipse to
    - create the Java code and WSDL description of the new Weather Service. The method is called "gibWetterberichteGebietAsString" and requires five parameters (boundaries as GPS latitude and longitude points, time). It returns a string that contains the prognosis for the given area.
2. Create the USDL description
    - The Service Engineer
    - uses
    - the USDL Editor
    - to add details about his service. E.g., WSDL-Import for creating technical functions, Capabilities, Provider, Price. This description includes
        - Short and long description
        - person (Michael Vanetti, Machabaer Str. 305, 50668 Koeln, Phone: +49-335511-1234432100) as provider
        - technical and functional interface description (method name, parameter names, access profile with URL)
        - price details (plan with flatrate for 1000€ per month, plan with 2€ per use, both with 19% VAT)
        - organization (V-Mind GmbH) as business owner, having Michael Vanetti as representative
        - target consumers (Customers of car insurance companies)
        - service levels (Authentication, 10ms service time)
3. Make the new service publicly available
    - The Service Engineer
    - uses the USDL Editor to upload
    - the USDL file and business logic to the USDL repository
    - to publish the service.

Craft Services

Owner: Andrea Horch & Maximilien Kintz, IAT der Universität Stuttgart

Type of use case: basic

Focus on Role: Service Provider and Service Consumer

Implementation status: Prototype already implemented

System Setup

- Web-Service Interface: Web Service Client
- Graphical User Interface: Web Browser

Roles

Service Provider, Service consumer

Activities

1. Register
    - Service Provider
    - Registration Interface (still to be implemented)
    - Craft Service Search Platform
    - Is registered in the Craft Services repository
2. Search
    - Service Consumer
    - Web Service Interface or GUI (Web browser)
    - Craft Service Search Platform
    - Gets list of matching craft services in USDL

Healthcare

Owner: Martin Schaeffler, Siemens AG

Type of use case: company internal use case

Focus on Role: Service Broker

Implementation status: planned (waiting for sap agora)

System Setup

- USDL-Editor M4 (Eclipse PlugIn)
- SAP Service Marketplace (agora)
- Use case description

Roles

- Service Broker(Insurance)
- Service Consumer
- Service Provider
- Service Marketplace

Activities

1. Offer services on the marketplace
    - Service Provider
    - describe and deploy service description
    - USDL Editor and marketplace repository
    - service is descripted and available on the marketplace
2. combine service on the marketplace
    - Service Broker (Insurance)
    - combine services to create new offerings for insurance customer
    - by using agregation tools
    - new services for insurance customer
3. inform client
    - Service Broker Insurance
    - inform insurance customer about special service offerings on the marketplace
    - by E-Mail
    - Service Consumer knows about the insurance offerings
4. Log on the marketplace
    - Service Consumer
    - with own user account
    - by using webbrower
    - Service Consumer is logged on the marketplace
5. Search for inscurance services
    - Service Consumer
    - filter and browse the service catalogue
    - by using the webbrowser
    - find insurance services
6. Change componentes of the selected service
    - Service Consumer
    - use the "select alternative componente" functionality in the catalogue
    - by using the browser
    - select the service components of his/her needs
7. Acknowledge service subscription
    - Service Consumer
    - use the "book/acknowledge" functionality in the marketplace
    - by using the browser
    - service booked and booking acknowledged
8. Inform service provider about changed subscriptions
    - Service Marketplace
    - sending booking information of the marketplace customer to the provider of the service
    - by E-Mail or system functionality

- ○ Service Provider is informed about the booking and has the customer data

IT Service Catalogue

Service Providers publish IT Services into the Service Catalog. Service Consumers browse/search the Service Catalog and subscribe to the Services. Thereafter they manage their Subscriptions.

Owner: Lukas Barton, Hewlett-Packard

Type of use case: basic

Focus on Role: Service Consumer

Implementation status: not planned yet

System Setup

- IT Service Catalog
- Web User Interface
- USDL Interface
- Web Browser

Roles

- Catalog Manager
- Service Provider
- Service Consumer

Activities

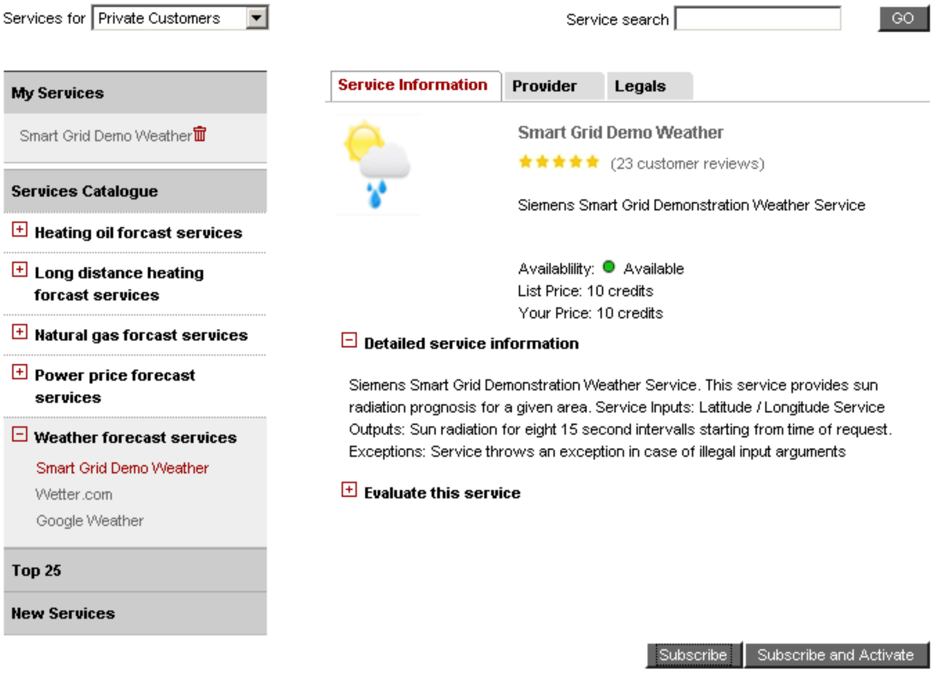
1. Create Catalog Structure
   - Catalog Manager
   - Web Browser
   - IT Service Catalog
   - Catalog Categories defined
2. Publish Service into Catalog
   - Service Provider
   - Web Browser
   - IT Service Catalog
   - Service published into Catalog
3. Browser and Search the Catalog
   - Service Consumer
4. Select Service and submit Service Order
   - Service Consumer
5. Deliver Service to Consumer
   - Service Provider
6. Manage Service Instance
   - Service Consumer
7. Rate Service
   - Service Consumer

1.5 Implementation of use cases

Energy Marketplace

In this scenario, services related to producing and consuming of energy, are offered on a service marketplace. The implementation was based on the Google Web Toolkit and tailored to the needs of a service marketplace for a Siemens energy marketplace i.e. adapting the user interface to the Siemens web appearance as well as designing an import mechanism for services described in USDL3M4. The marketplace prototype is used for demonstration purposes.

After the manual import of the USDL file, the information is stored in the service repository and is accessible for the marketplace customer via the Service Catalogue in the user interface. With the import, the provider could choose or define topic cluster in which the services are organized. As the customer navigates to a service of interest in the catalog details of this service are displayed. The overall and coarse-granular description, availability and the main significant price model are displayed in the “Service Information” tab of the detail side and the user can navigate to other details by selecting the tabs “Provider” and “Legal”. The roles “administrator”, “provider”, and “customer” are implemented to use the marketplace for demonstration purposes. The screenshot depicts the main screen for the customer of the service marketplace.



Smart Grid Demo Weather

- ⊞ Heating oil forcast services
- ⊞ Long distance heating forcast services
- ⊞ Natural gas forcast services
- ⊞ Power price forecast services
- ⊟ Weather forecast services
  - Smart Grid Demo Weather
  - Wetter.com
  - Google Weather

Smart Grid Demo Weather

★★★★★ (23 customer reviews)

Siemens Smart Grid Demonstration Weather Service

Availability: ● Available
List Price: 10 credits
Your Price: 10 credits

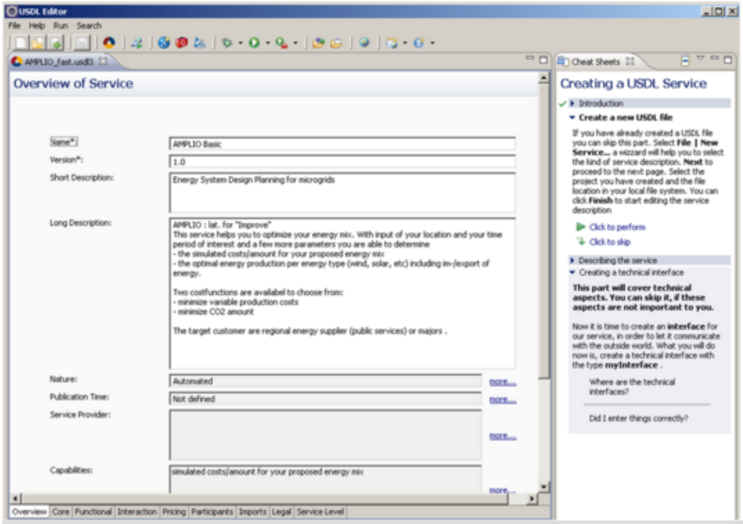
⊟ Detailed service information

Siemens Smart Grid Demonstration Weather Service. This service provides sun radiation prognosis for a given area. Service Inputs: Latitude / Longitude Service Outputs: Sun radiation for eight 15 second intervalls starting from time of request. Exceptions: Service throws an exception in case of illegal input arguments

⊞ Evaluate this service

Figure 2: The energy marketplace prototype

The implementation of a second scenario of the Smart Grid context of the energy domain bases on tooling provided by SAP. For technical details please refer to the SAP description. The tooling consists of the USDL editor, the USDL repository and the USDL marketplace. The screenshots depict the tooling for this implementation. The service provider describes his services by help of the USDL editor and could upload the output to the USDL repository. The repository could be connected to one or many marketplaces with support USDL. In our scenario, one marketplace is connected and the USDL file form the repository is published on it. The provider is then able to adjust some marketplace-specific parameters and set the service offering as active. The service consumer searches for services and could contact the provider to get more information, negotiate the conditions or book the service as it is.



USDL Editor
File Help Run Search
AMPLIO_fast.usdl3
Cheat Sheets
Overview Core Functional Interaction Pricing Participants Imports Legal Service Level
more...

Overview of Service

Name*: AMPLIO Basic
Version*: 1.0
Short Description: Energy System Design Planning for microgrids
Long Description:
AMPLIO : lat. for "Improve"
This service helps you to optimize your energy mix. With input of your location and your time period of interest and a few more parameters you are able to determine
- the simulated costs/amount for your proposed energy mix
- the optimal energy production per energy type (wind, solar, etc) including im-/export of energy.

Two costfunctions are availabel to choose from:
- minimize variable production costs
- minimize CO2 amount

The target customer are regional energy supplier (public services) or majors .

Nature: Automated
Publication Time: Not defined
Service Provider:
Capabilities: simulated costs/amount for your proposed energy mix

Creating a USDL Service

✓ Introduction

Create a new USDL file

If you have already created a USDL file you can skip this part. Select **File | New Service...** a wizzard will help you to select the kind of service description. **Next** to proceed to the next page. Select the project you have created and the file location in your local file system. You can click **Finish** to start editing the service description

▶ Click to perform
Click to skip

Describing the service

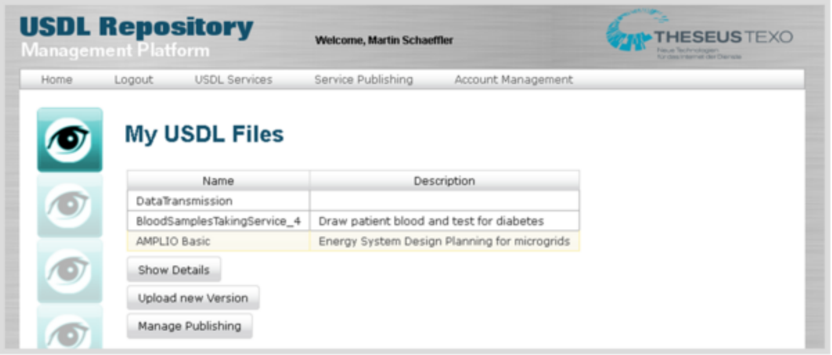
Creating a technical interface

This part will cover technical aspects. You can skip it, if these aspects are not important to you.

Now it is time to create an **interface** for our service, in order to let it communicate with the outside world. What you will do now is, create a technical interface with the type **myInterface** .

Where are the technical interfaces?

Did I enter things correctly?



Welcome, Martin Schaeffler

My USDL Files

Name	Description
DataTransmission	
BloodSamplesTakingService_4	Draw patient blood and test for diabetes
AMPLIO Basic	Energy System Design Planning for microgrids

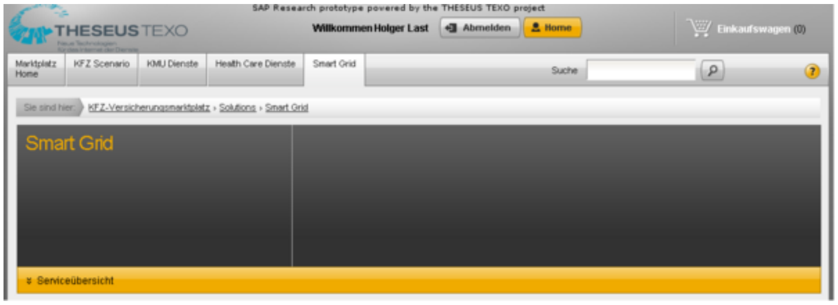


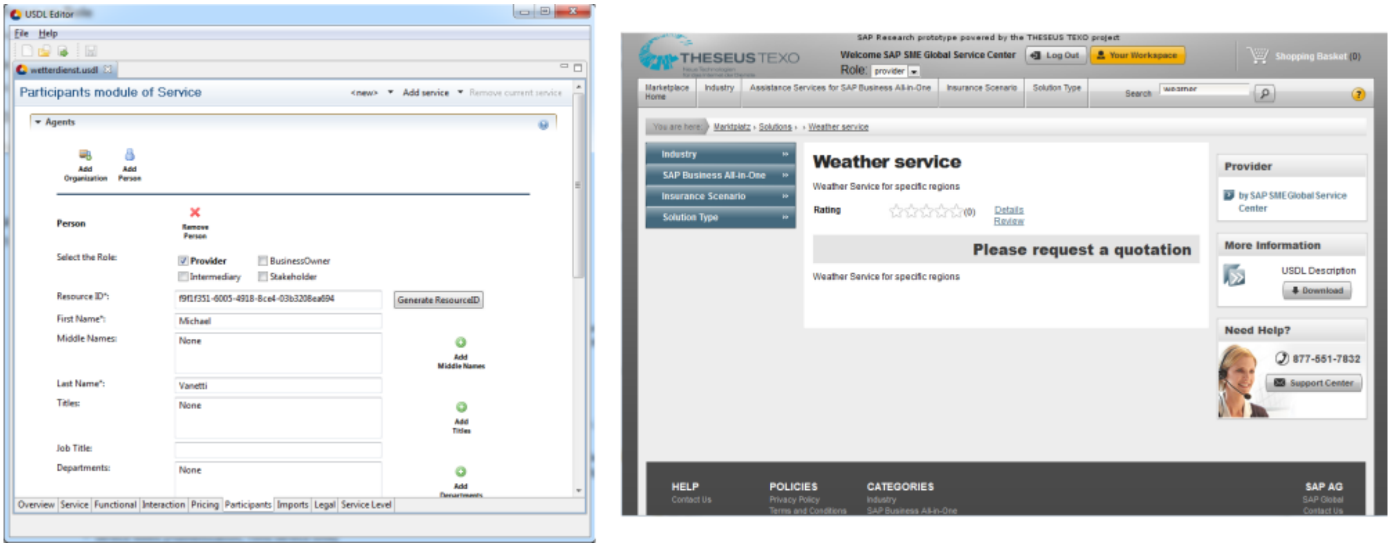
Figure 3-5: USDL Editor, Repository and Marketplace in the Siemens prototype

Creating and uploading a service

This prototype consists of two components:

- The USDL Editor. It is used by the Service Provider to create the USDL description. Depending on the actual service, it can be complemented by something to create actual business logic, like Eclipse for implementing some java classes, BPMN workflows and the like. The editor is available as open source software from Sourceforge.net (download). It is based on Eclipse and is technically an Eclipse Rich Client Platform tool.
- The USDL Marketplace. Services will be registered at the marketplace, which extracts relevant information out of USDL and indexes them. Afterwards, they can be searched, found, and compared via a typical marketplace frontend. The marketplace was developed using JBoss and JBoss seam, Postgresql database and some other tools.

The presented integration is based on USDL 3M5, which is the latest version of the language. The screenshot shows the two tools in action.


Participants module of Service

Person

Select the Role: ☑ Provider ☐ BusinessOwner ☐ Intermediary ☐ Stakeholder

Resource ID*: f9f1f351-6005-4918-8ce4-03b3208ea694

First Name*: Michael

Middle Names: None

Last Name*: Vanetti

Titles: None

Job Title:

Departments: None

Weather service

Weather Service for specific regions

Rating ☆☆☆☆☆ (0)

Please request a quotation

Weather Service for specific regions

Provider: by SAP SME Global Service Center

More Information: USDL Description

Need Help? 877-551-7832

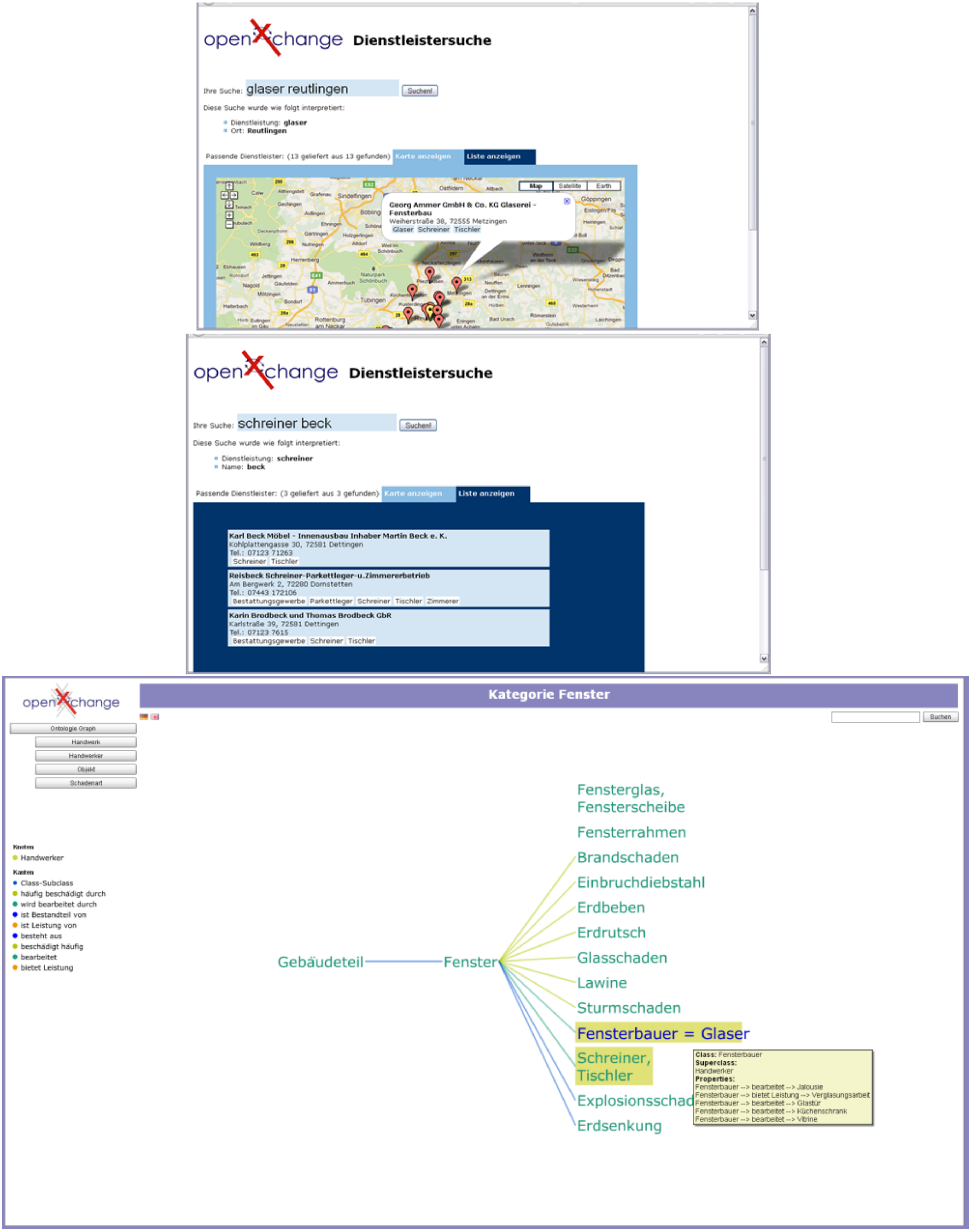
Figure 3: USDL Editor and Marketplace from SAP

Craft Services

The prototype for craft services search consists of three components:

- A SOAP Web service used to perform semantic search queries in a craft services database. The requests are sent in a custom XML format, the answers (i.e. the description of the matching craft services) are sent using USDL.
- An HTML-based search interface used to query the Web service in a user friendly way. See the following two screenshots: Screen-Map and Screen-List.
- An ontology browser used to navigate through the German Crafts and Craftsmen Ontology (GeCCO), which allows user friendly exploration of the complex semantic classification of crafts. The ontology is directly linked to the HTML-based search interface (clicking on a craft name starts the corresponding search requests). See screenshot: Screen-OntoBrowser.

The Web interface and the search Web service run on a Tomcat Server, we use a MySQL database as our repository. The ontology browser uses the Prefuse Flare visualization framework and the GeCCO ontology modeled in OWL.



openXchange Dienstleistersuche

Ihre Suche: glaser reutlingen

Diese Suche wurde wie folgt interpretiert:

- Dienstleistung: **glaser**
- Ort: **Reutlingen**

Passende Dienstleister: (13 geliefert aus 13 gefunden)

Georg Ammer GmbH & Co. KG Glaserei - Fensterbau
Weiherstraße 38, 72555 Metzingen
Glaser Schreiner Tischler

openXchange Dienstleistersuche

Ihre Suche: schreiner beck

Diese Suche wurde wie folgt interpretiert:

- Dienstleistung: **schreiner**
- Name: **beck**

Passende Dienstleister: (3 geliefert aus 3 gefunden)

Karl Beck Möbel - Innenausbau Inhaber Martin Beck e. K.
Kohlplattengasse 30, 72581 Dettingen
Tel.: 07123 71263
Schreiner Tischler

Reisbeck Schreiner-Parkettleger-u.Zimmererbetrieb
Am Bergwerk 2, 72280 Dornstetten
Tel.: 07443 172106
Bestattungsgewerbe Parkettleger Schreiner Tischler Zimmerer

Karin Brodbeck und Thomas Brodbeck GbR
Karlstraße 39, 72581 Dettingen
Tel.: 07123 7615
Bestattungsgewerbe Schreiner Tischler

openXchange

Kategorie Fenster

Knoten
- 🟡 Handwerker

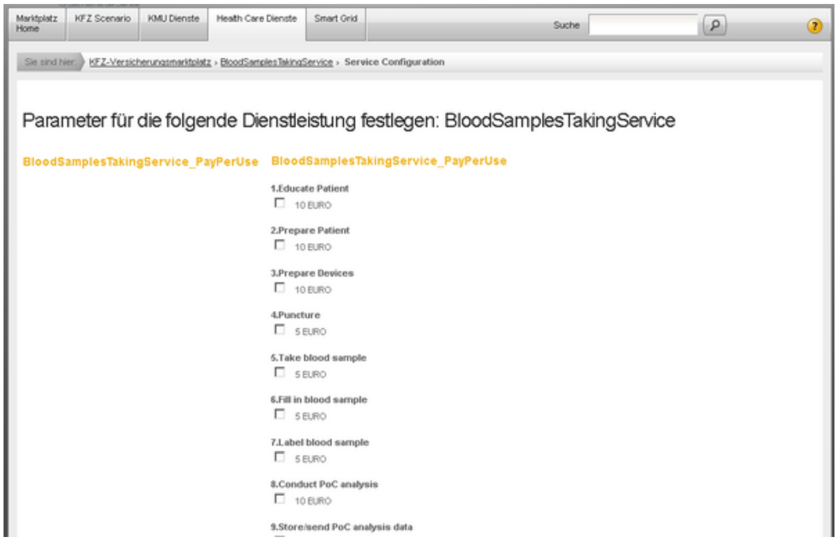
Kanten
- 🔵 Class-Subclass
- 🟢 häufig beschädigt durch
- 🟢 wird bearbeitet durch
- 🔵 ist Bestandteil von
- 🟠 ist Leistung von
- 🔵 besteht aus
- 🟢 beschädigt häufig
- 🟢 bearbeitet
- 🟠 bietet Leistung

Class: Fensterbauer
Superclass:
Handwerker
Properties:
Fensterbauer --> bearbeitet --> Jalousie
Fensterbauer --> bietet Leistung --> Verglasungsarbeit
Fensterbauer --> bearbeitet --> Glastür
Fensterbauer --> bearbeitet --> Küchenschrank
Fensterbauer --> bearbeitet --> Vitrine

Figure 6-8: Screenshots of Map view, List view and Ontology browser of University of Stuttgart's prototype

Healthcare

The implementation of the healthcare scenario bases on tooling provided by SAP. For technical details please refer to the SAP description. The tooling consists of the USDL editor, the USDL repository and the USDL marketplace. The procedure for creating and publishing the service is comparable to the energy implementation II: The service provider describes his services by help of the USDL editor and could upload the output to the USDL repository. The repository could be connected to one or many marketplaces with support USDL. In our scenario, one marketplace is connected and the USDL file form the repository is published on it. The provider is then able to adjust some marketplace-specific parameters and set the service offering as active. The service consumer searches for services and could contact the provider to get more information, negotiate the conditions or book the service as it is. In contrast to the energy implementation II, the customer is able to configure his services to his needs and contact the service provider with this detailed request. The screenshot shows the configuration of the service by the customer on the service marketplace. The tooling is not yet ready to support a service aggregator role. But the configuration of the service is a first approach to customize services for the needs of the customer. The prosecution of the implementation of the test case is planned for the time after the tooling is released to open source and could be adapted for this test case.


Marktplatz Home
KFZ Scenario
KMU Dienste
Health Care Dienste
Smart Grid
Suche
?
Sie sind hier: KFZ-Versicherungsmarktplatz › BloodSamplesTakingService › Service Configuration

Parameter für die folgende Dienstleistung festlegen: BloodSamplesTakingService

BloodSamplesTakingService_PayPerUse

BloodSamplesTakingService_PayPerUse

1.Educate Patient
□ 10 EURO

2.Prepare Patient
□ 10 EURO

3.Prepare Devices
□ 10 EURO

4.Puncture
□ 5 EURO

5.Take blood sample
□ 5 EURO

6.Fill in blood sample
□ 5 EURO

7.Label blood sample
□ 5 EURO

8.Conduct PoC analysis
□ 10 EURO

9.Store/send PoC analysis data

Figure 9: Configuration of the service on the Siemens Service Marketplace

2 What is the outcome?

In addition to the completely revised USDL version 3.0, reflected in milestone M5, there have been several additional outcomes of the incubator group. First and foremost, feedback has been collected for future milestones of USDL which is presented in the following. Second, the incubator group's partners investigated and/or applied USDL in their setting. Further below the partners assessments are given accordingly.

2.1 Feedback for next iteration

One particular outcome of the incubator group is reviews and suggested improvements by partners. First, founding member attensity provided an in-depth review. Second, incubator member DERI suggested the inclusion of service offerings as a new concept in future USDL versions. Third, there was additional feedback with minor points concerning several USDL M5 modules. All points are documented in detail at D2.3.

A further outcome besides the actual specification concerns methods for USDL. Besides the meta-model of USDL guidelines for extensions and protocols are required for a successful standard. Several methods which have been elaborated in the course of the incubator group are discussed below. In addition, a linked data representation of USDL has been proposed in order to better align with the W3C Semantic Web Activity.

2.2 Variant management

Different variants of USDL are required for different contexts. This is already shown by the Legal Module which requires different contents depending on the jurisdiction of a country. The issue aggravates if more and more parameters are relevant to determine the correct variant. Therefore, the incubator group investigated one possible solution for variant management consisting of a canonical data model, a context driver mechanism, governance processes, and appropriate tooling. Although the solution for variant management is targeted at existing business documents, such as a purchase order, it provides a powerful and adequate means for dealing with USDL variants as well. More information can be found at variant management.

2.3 Protocol for accessing service repository

Another methodological aspect of USDL concerns a standardized access protocol to arbitrary service repositories on the basis of USDL. Such a standardized protocol would facilitate building tools, such as editors, to allow access to arbitrary repositories.

A particular proposal for such a protocol has been made by HP during the course of the incubator group. Because REST and Atom Publishing Protocol can be regarded de-facto standards it was recommended to base the protocol on those specifications.

We should define Atom Binding - an API mapping to Atom which supports create, read, update, delete and query operations with a USDL enabled Repository. It should be designed to maximize the information available to an Atom reader while still supporting the meta-data richness available in the USDL data model. USDL entities like Services or Service Bundles could be mapped onto collections - i.e. feed of entries with URL similar to /services?order-by={field1,field2,..}&page-size={page_size}&start-index={start_index}&p.{property_name}={property_value}&inline-content.

We can use following mapping:

- atom:id - resourceID
- atom:title - name
- atom:summary - description (if present)
- atom:updated
- atom:published - publicationTime
- atom:author - provider
- atom:category - classifications
- atom:content – already standardized XML serialization of entity

2.5 Representation in LoD

Recently, the so-called Linked Data principles were presented as a means towards creating a Web of Data better suited for machine processing. These principles establish that one should:

1. Use URIs as names for things,
2. Use HTTP URIs so that people can look up those names,
3. Provide useful information, using the standards (RDF*, SPARQL) when some-one looks up a URI,
4. Include links to other URIs so that they can discover more things.

Since these principles were proposed we have witnessed an outstanding growth in terms of data and vocabularies allowing people to freely expose and interlink large quantities of heterogeneous data. In fact, for raw data that can effectively be modelled in RDF, Linked Data principles are nowadays considered as the best means for publishing it on the Web.

In a nutshell, the first principle ensures that resources are uniquely identified. The second principle ensures that their identification is such that HTTP can be used for obtaining information about the resource. The third principle establishes standard technologies for exposing data in a manner that is suitable for machine processing. Finally, the fourth principle aims to promote the interlinking of data. The same way hyperlinks connect Web documents into a single global information space, Linked Data uses hyperlinks to connect data into a single global data space. These links allow applications to navigate the Web of Data and, since the data is exposed through HTTP (see principle 2) and represented in some standard format (see principle 3), machines can obtain it, interpret it, and act accordingly in an automated manner.

USDL was originally modelled in eCore and integrates a number of different per- spectives over services (e.g., organisations, Web services, resources, etc). However given that the main goal is to turn it into a global scheme for describing, exposing and trading services on a large scale, Linked Data principles appear to be most ap- propriate for publishing USDL data on the Web. In the reminder of this chapter we shall thus focus on how USDL can be adopt linked data principles, focusing notably on the representation of USDL in RDF(S), and on the interlinking with external vocabularies and data sources.

A linked data vocabulary for USDL

To create a linked data version of USDL, requires to model USDL in RDF(S). Covering all modules of USDL would lead to revisiting many aspects of the semantics of USDL. Instead we shall focus herein in the main design decisions concerning the lightweight semantic representation of USDL in RDF(S)/OWL, and those targeted at promot- ing and better supporting the exposition of USDL information on the Web of Data through the reuse existing vocabularies and instances. Through this exercise it shall be seen not only that existing vocabularies cover a good part of USDL, but it shall also show that modelling USDL in this manner has a number of benefits like the support for additional capabilities and the compatibility with existing datasets.

The forth Linked Data principle is to include links to other data sources from the Web. These links are an essential means towards generating a Web of Data as opposed to disconnected silos as they simplify data integration and interpretation, as well as they enable the discovery of additional data in what is often referred to as a follow-your-nose fashion. A fundamental activity in adapting USDL for the Web of Data therefore concerns the analysis of existing vocabularies and datasets, in order to i) identify reusable vocabularies to avoid reinventing the wheel and promote reuse and integration; and ii) identify possible relationships with USDL concepts and USDL data to support navigation across datasets and to simplify data integration.

The review of USDL according existing data sets lead to the recommendation of the following vocabularies:

- ctag: <http://commontag.org/ns#>
- dcterms <http://purl.org/dc/terms/>
- dctypes: <http://purl.org/dc/dcmitype/>
- foaf: <http://xmlns.com/foaf/0.1/>
- GoodRelations: <http://purl.org/goodrelations/v1#>
- Minimal Service Model: <http://cms-wg.sti2.org/ns/minimal-service-model#>
- org: <http://www.w3.org/ns/org#>
- skos: <http://www.w3.org/2004/02/skos/core#>
- time: <http://www.w3.org/2006/time#>
- vcard: <http://www.w3.org/2006/vcard/ns#>
- other vocabluaries like e3service, SimplePartWhole (W3C), Geonames, WGS84, or W3C Geo XG might further be considered.

A USDL core vocabulary

We further recommend to build a lightweight USDL core vocabulary. In this respect we suggest to remove or simplyfy some of the USDL modules.

1. The legal and pricing module should be separate vocabularies, because they address generic properties and can be used also beyond describing USDL services.
2. the SLA module also can be separated from the USDL core, which enable to use different vocabularies to describe the service level aspects of services depending on the application domain and requirements.
3. The break-down of a service into several functions (USDL function module) can also be achieved by describing the service in terms of service composition and a set of powerful relations between services. For the sake of simplicity the functional module can

therefore be dropped in favor for a light-weight core vocabulary.

4. Using the Minimal Service Model vocabulary (msn) as a replacement for the USDL technical interface module, so the technical module can be dropped completely. Instead of linking the technical interaface to USDL functions (which were dropped in favor of sub-services), we suggest to link it to the service interaction. So the technical interface becomes a msm:Service which is linked to the usdl:Interaction.

The following picture shows a possible core USDL service vocabulary according to the recommendations above, including the linkage of other existing vocabularies.

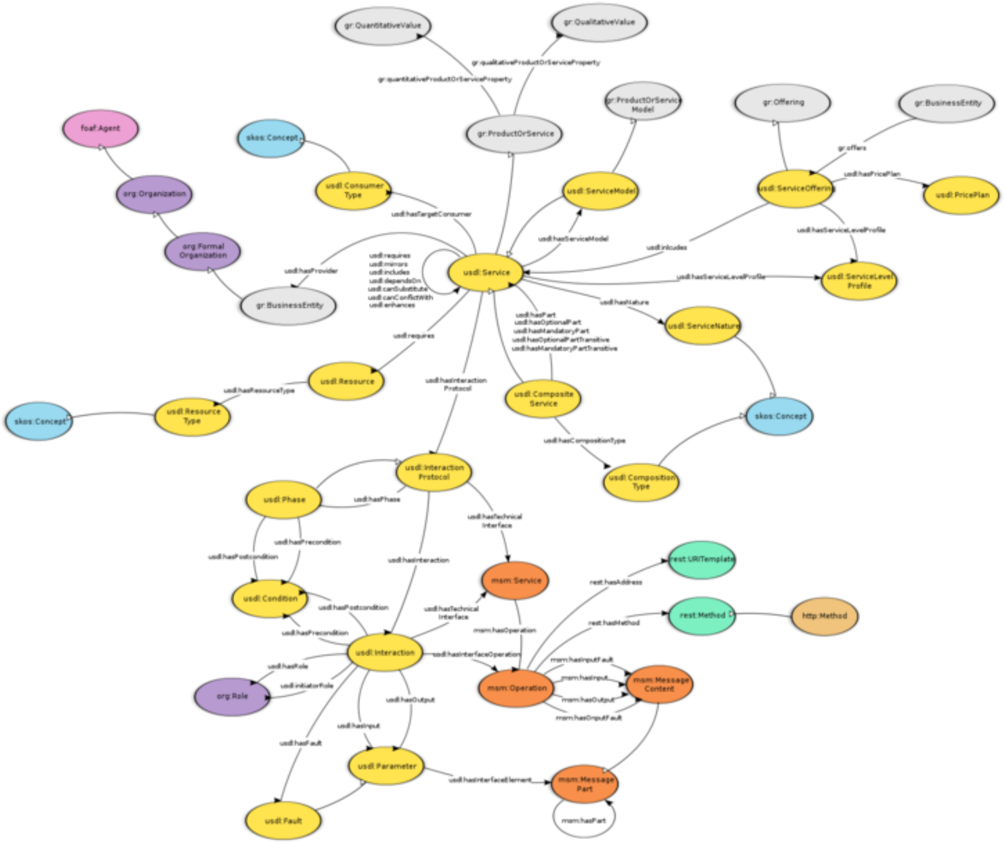


Figure 10: USDL core vocabulary in Linked Data

2.6 Assessment per partner

Before closing the XG, members and invited experts were asked for their opinion about USDL and its potential.

XG Members

HP

We can use USDL as another mean of exposing data stored in our IT Service Catalog. We are able to map most of the data stored in the Catalog. Although in some areas we would need to expose more information (e.g. pricing model, service configuration).

DERI Galway

USDL provides a modular, flexible and quite complete model/language for describing services from both business and IT perspectives. USDL builds on lessons learned from previous efforts such as OWL-S and WSMO. From that perspective, USDL presents two unique features. First, it covers service aspects quite exhaustively. Second, USDL focused only on describing services and their aspects without assuming/interfering with a specific modelling paradigm.

Like any description language, USDL has to be extensively validated in real use cases. One potential issue where USDL can deal with assuming some extensions is describing highly configurable services. This kind of services cannot be described statically since some of their parameters are request-dependent, dynamic or correspond to sensitive information. Discovery and selection of highly configurable services is particularly important for enabling late service binding and making service offers truly comparable.

The Open University

USDL represents a comprehensive description for services from a business point of view based on the experiences of several key industrial players and large service description initiatives including WSMO and OWL-S. In our view the key to the take-up of USDL will be to extent to which USDL descriptions can be shared between interested parties and linked to other descriptions, standards and formats. Linked Data is proving to be a successful technology for integrating data at scale and thus we would wish to see a continuation of the work in representing USDL within RDF(S) and also linking to existing Linded Data vocabularies.

Fraunhofer Gesellschaft

We needed a service description language since the goal of our Web service was to return lists of matching craft services. USDL proved to be an extensive yet relatively simple service description language and could be easily integrated. Describing our search Web service using USDL was a bit more tricky and required some training. Overall, we found USDL an extensive, relatively complex but usable service description language.

Universidad Polytecnica de Madrid

At UPM we consider that the modular approach followed in this USDL specification has been a right decision. In fact, we have succesfully used similar approaches when describing user generated services. Separating the different service aspects into independent facets specified in their own language has proved to be efficent, flexible and easy to be processed by service delivery platforms. Furthermore, we foresee that USDL can be directly used in the new mashup and widget environments, as a common format among the mashup creation environments, mashup runtimes and with user devices (browsers, smartphones, TVs, cars, etc).

Flexibility is probably the most important feature we find in USDL, since it makes very easy to adapt USDL specification to new requirements that may be needed in the future. We would also encounter very interesting the possibility to apply the USDL specification to the scope of Telco services, thus opening a new scope of potential application fields such as the seamless integration of telco and IT services. In this last sense, we are investigating the potential application of USDL in a telco mashup platform developed in the context of FP7 project Omelette.

For the future it would also be quite interesting to include somewhere any reference to identity and privacy concerns in order to support techniques of identity and privacy management that may be applied in service delivery platforms. It can be even more important due to the legislation that regulates this in different countries. These aspects might be included in what is currently called legal module, although we see that this module is more related to rights and obligations mainly reffered to licensing issues.

University of Helsinki

Efficient operation of organizations in service ecosystems requires business-IT alignment within organizations as well as means for enabling inter-organizational interoperability. USDL addresses these issues by providing a language for describing business services in a holistic and unified manner, which considers both business and technological aspects of ICT-enabled services. Another strength of the USDL -approach is the underlying assumption that one language does not fit all usage scenarios. Thus a mechanism for managing variability of service descriptions is outlined in the USDL. The future USDL development planned during the incubator group addresses issues related to discovery, combination and management of service knowledge with Linked Open Data (LOD) -approach. This aligns well with the current movements in organizations and governmental institutions for enabling more efficient collaboration in service ecosystems with LOD -approaches. Overall, I see the current and planned development of the USDL as an essential effort towards enabling sustainable operation of organizations in future service ecosystems.

Invited experts

Konstadinos Kutsikos

Our first encounter with USDL (and competing efforts) was during the final stage of the SYNERGY FP7 project, which was focused on cross-enterprise development and use of knowledge-intensive services. Beyond the need for describing technical aspects of such services in a homogeneous way, our overall requirement was for a flexible/modular service description language for capturing the business, operational and technical service parameters.

In this context, we selected USDL for exploring the requirements that new business models for eServices may impose on service description mechanisms, focusing in particular on variant management (i.e. adjusting service description for different context). Beyond USDL's legal module, I believe that adopting variant management on the Pricing module could help consider enriched pricing and billing mechanisms. When dealing with knowledge-intensive services (an increasingly large share of high-value eServices) that involve co-production of intellectual property, the pricing structure will inevitably be sophisticated. For example, the commercial exploitation of intellectual property may require that both service description and billing mechanisms account for shared revenue arrangements. Or, that they account for financial instruments that relate to future consumption of services (e.g. financial options).

Overall, I believe that USDL has great potential in the market and the host of companies and people behind it will prove its commercial value.

Initiating members

SAP

The systematic use of the Internet for new ways of value creation in the services sector (frequently dubbed Web service ecosystems, Internet of Services, or simply Future Internet) requires a standard that creates a “commercial envelope” around a service. We consider USDL to be one of the foundational technologies for setting up such a Future Internet around today's core enterprise systems. Consequently, we would like to see the standardization of USDL to be continued after its incubation.

Siemens

Services, service marketplaces and service ecosystems are getting more and more important in Siemens business. The USDL approach to combine descriptions for business and IT just simplifies the provision, comparison, use and delivery of value-added services and therefore supports this development and the widespread use. Therefore we support the continuation of the standardization of USDL under the W3C.

Attensity

Today, when searching and evaluating services, we are still confronted with fragmented and incomplete service descriptions. Some formalism offer rich technical descriptions but lack legal descriptions or vice versa. USDL is a great opportunity to overcome this situation by unifying the different representations and by providing guidance for how and what to describe to the service providers. For service consumers it will facilitate the comparability of services and hence enable an informed selection of services. As a software vendor, Attensity therefore views USDL as an important market enabler for an active service market. Furthermore, to ensure that this becomes an open and accessible market we strongly recommend and actively support the standardization effort for USDL within the W3C.

DFKI

One vision underlying the IOS (Internet Of Services) is that the realization of a comprehensive web services infrastructure will boost the market creating a new efficient a promising marketplace that will integrate in the standard business services economy. Past efforts on web services have been concentrated only on technical realization of services. This still did not provide reliable solutions for e.g., services discovery, delivery (privacy, timelines, reliability) which are crucial for a broad adoption of the technology by the traditional physical service providers. USDL responds to the need of filling the gap between the business, operational and technical perspectives of services by providing common and unified service semantics. This will simplify cross-enterprise integration providing a trading infrastructure for the IOS (internet of services). As this semantics has to be generally accepted we think that the standardization of USDL is a crucial step forward in the development of the IOS.

3 What is the conclusion?

We conclude the Incubator Group by formulating the following recommendations:

1. **Recommendation**: Having a language specification is not everything. To support a wide range of instances, interoperability and implementations, some methods or defined processes are necessary. The legal module, for instance, will strongly differ according to the country in which the service is located. This raises the demand for systematic support of variants of that language. Adoption of USDL would also benefit from a query language for accessing USDL documents on some repository or even information within these USDL documents.
2. **Recommendation**: Since the work on USDL is not finished (see recommendation 1), we recommend that the work of formally specifying USDL also under consideration of existing standards is continued at W3C. The use cases show that there are fields of application for USDL. The number of registered participants in the XG foster this (15 organizations, 38 persons, 4 invited experts, plus ~15 persons in the public mailinglist)
3. **Recommendation**: Due to the nature of USDL, being more or less the glue between different description formats and languages while adding some own concepts, it's recommendable that it receives a tighter integration with the Semantic Web in general and Linked Data in particular.
4. **Recommendation**: Again, due to it's motivation of describing services, we recommend an integration with effort that is located in the Web Service Activity.

References

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

Akkermans et al. (2004) Value Webs: Ontology-Based Bundling of Real-World Services. IEEE Intelligent Systems, vol. 19 (4) pp. 57-66

Alter, S. (2008) Service system fundamentals: Work system, value chain, and life cycle. IBM Systems Journal 2008, 47(1), 71-85.

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S. & Xu, M. (2007) Web Services Agreement Specification (WS-Agreement). Open Grid Forum, http://www.ogf.org/documents/GFD.107.pdf

Ankolekar, A., Burstein, M. H., Hobbs, J. R., Lassila, O., Martin, D. L., McIlraith, S. A., Narayanan, S., Paolucci, M., Payne, T. R., Sycara, K. P. & Zeng, H. (2001) DAML-S: Semantic Markup for Web Services. In: Proceeding of the first Semantic Web Working Symposium (SWWS), Stanford University, California, pp. 411-430,

Baida, Z., Gordijn, J. & Akkermans, H. (2001) Service Ontology. Free University Amsterdam Barros, A. & Dumas, M. (2006) The Rise of Web Service Ecosystems. IT Professional, 8:5, 31-37.

Baumann, C. & Loes, C. (2010) Formalizing Copyright for the Internet of Services. In: Proceedings of the 1st International Workshop on the Internet of Services, pp. (to appear), ACM Digital Library

Berre, A. J. (2008) UPMS - UML Profile and Metamodel for Services - an Emerging Standard. In: Proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference, Munich, IEEE Press

Berre, A. J. (2009) Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS). OMG Document. http://www.omg.org/spec/SoaML/1.0/Beta2/

Bitsaki, M., Danylevych, O., van den Heuvel, W., Koutras, G., Leymann, F., Mancioppi, M., Nikolaou, C. & Papazoglou, M. (2008) An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network. In: LNCS An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network, pp., Springer Berlin

Buxmann, P., Hess, T. & Ruggaber, R. (2009) Internet der Dienste. In: Wirtschaftsinfomatik 5 (2009), pp. 341,342

Dhanesha, K. A., Hartman, A. & Jain, A. N. (2009) A Model for Designing Generic Services. In: Proceedings of the Seventh IEEE International Conference on Services Computing, pp. 435-442, IEEE Press

DIN PAS 1018 (2002) Grundstruktur für die Beschreibung von Dienstleistungen in der Aus-schreibungsphase, Berlin – German standard for the description of services for tendering. Beuth Verlag GmbH, Ref. Nr. PAS 1018 : 2002-12, Vertr.-Nr. 29019

Dobson, G. & Sánchez-Macián, A. (2006) Towards Unified QoS/SLA Ontologies. In: Proceedings of the IEEE Services Computing Workshops (SCW 2006), Third International Semantic and Dynamic Web Processes Workshop (SDWP 2006), pp. 169-174, IEEE Press.

Emmrich, A. (2005) Ein Beitrag zur systematischen Entwicklung produktorientierter Dienstleistungen, Dissertation, Wirtschaftsinformatik, vol 170, HNI-Verlagsschriftenreihe, Paderborn, Heinz Nixdorf Institut, Universität Paderborn, 2005, ISBN 3-935433-79-4.

Estefan, J. A., Laskey, K., McCabe, F. G. & Thornton, D. (2009) Reference Architecture Foundation for Service Oriented Architecture. Version 1.0, Committee Draft 2, October 2009, OASIS Service Oriented Architecture Reference Model TC

Farrell, J. & Lausen, H. (2007) Semantic Annotations for WSDL and XML Schema (SAWSDL), W3C Recommendation, August 2007, http://www.w3.org/TR/sawsdl/

Fensel, D., Fischer, F., Kopecký, J., Krummenacher, R., Lambert, D., Vitvar, T. (2010) WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web, W3C Member Submission 23 August 2010 http://www.w3.org/Submission/WSMO-Lite/

Ferrario, R. & Guarino, N. (2008) Towards an Ontological Foundation for Services Science. In: Proceedings of Future Internet Symposium 2008, pp. Springer Verlag, Berlin

Gangadharan, G. R., D'Andrea, V., Iannella, R. & Weiss, M. (2007) ODRL Service Licensing Profile (ODRL-S). In: Proceedings of the 5th Intl. Workshop for Technical, Economic, and Legal Aspects of Business Models for Virtual Goods, pp. 1–17, .

Gomadam, K., Ranabahu, A. & Sheth, A. (2010) SA-REST: Semantic Annotation of Web Resources, W3C Member Submission, April 2010, http://www.w3.org/Submission/SA-REST/

Gordijn, J., Akkermans, H. (2001): Designing and Evaluating e-Business Models. IEEE Intelligent Systems, vol. 16 (4) pp. 11-17

Harding, C. (2008) Service-Oriented Architecture Ontology. The Open Group Draft 2.0, http://www.opengroup.org/projects/soa-ontology/ . Accessed 2009-06-15

Kelkar, O., Leukel, J. & Schmitz, V. (2002) Price modeling in standards for electronic product catalogs based on XML. In: Proceedings of the 11th World Wide Web Conference (WWW), pp. 366-375.

Kiemes, T., Oberle, D. & Novelli, F. (2010) Generic Modeling and Management of Price Plans in the Internet of Services. In: Workshop "Internet der Dienste", September 2010, GI-Jahrestagung Informatik 2010.

De Kinderen, S. & Gordijn, J. (2008a) e3Service - A model-based approach for generating needs-driven e-service bundles in a networked enterprise. In: Proceedings of 16th European Conference on Information Systems,

De Kinderen, S. & Gordijn, J. (2008b) e^3Service - An ontological approach for deriving multi-supplier IT-service bundles from consumer needs. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, pp. 318(1-10), IEEE Press.

Hadley, M. (2009) Web Application Description Language (WADL), W3C Member Submission, 31 August 2009, http://www.w3.org/Submission/wadl/

Kopecky, Jacek and Gomadam, Karthik and Vitvar, Tomas (2008): hRESTS: an HTML Microformat for Describing RESTful Web Services. In: The 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI2008), November, IEEE CS Press, Sydney, Australia

Kopecky, Jacek and Vitvar, Tomas and Pedrinaci, Carlos and Maleshkova, Maria (2011). RESTful Services with Lightweight Machine-readable Descriptions and Semantic Annotations. In Wilde, Erik and Pautasso, Cesare (eds.) REST: From Research to Practice, Springer.

Lehmann, S. & Buxmann, P. (2009) Pricing Strategies of Software Vendors. Business & Information Systems Engineering, 1(6), 452-462.

Loutas, N.; Peristeras, V.;Tarabanis, K. (2011) Towards a reference service model for the Web of Services, Data and Knowledge Engineering (2011)

MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F. & Metz, R. (2006) Reference Model for Service Oriented Architecture 1.0, Official OASIS Standard, October 2006, http://www.oasis-open.org/committees/soa-rm/

McIlraith, S. A., Son, T. C. & Zeng, H. (2001) Semantic Web Services. IEEE Intelligent Systems, 16(2), 46-53 de Miranda, J. G. B. & Baida, Z. (2006) Modelling pricing for configuring e-service bundles. In: BLED 2006 Proceedings.

Nagle, T. T. & Hogan, J. E. (2005) The Strategies and Tactics of Pricing. 4th edn. Prentice Hall.

Nagle, T. T. & Hogan, J. E. (2006) Segmented Pricing: Using Price Fences to Segment Markets and Capture Value. SPG Insights.

Nayak, N., Flaxer, D., Huang, Y., Marston, D., Nigam, A. & Sanz, J. (2010) A Unified Service Model for Service-oriented Business. IBM T. J. Watson Research Center, IBM Confidential Report.

Norton, B., Kerrigan, M., Mocan, A., Carenini, A., Cimpian, E., Haines, M., Scicluna, J. & Zaremba, M. (2008) Reference Ontology for Semantic Service Oriented Architectures. OASIS Public Review Draft 0.1, November 2008, http://docs.oasis-open.org/semantic-ex/ro-soa/v1.0/pr01/see-rosoa-v1.0-pr01.html

Oaks, P., Hofstede, A. H. M. & Edmond, D. (2003) Capabilities: describing what services can do, In: Proceedings of the International Conference on Service-Oriented Computing (ICSOC), pp. 1-16, LNCS 2910, Springer.

Oberle, D., Bhatti, N., Brockmans, S., Niemann, M. & Janiesch, C. (2009) Countering Service Information Challenges in the Internet of Services, Journal of Business & Information System Engineering (BISE), 2009, 1(5), 370-390

Oberle, D., Lamparter, S., Grimm, S., Vrandecic, D., Staab, S. & Gangemi, A. (2006) Towards ontologies for formalizing modularization and communication in large software systems. Applied Ontology 1(2), pp. 163-202, IOS Press.

O'Sullivan, J. (2006) Towards a Precise Understanding of Service Properties. Dissertation, Faculty of Information Technology, Queensland University of Technology, pp. 1-232.

Pandit, B., Popescu, V. & Smith, V. (2009) SML Service Modeling Language, V1.1. W3C Recommendation, May 2009. http://www.w3.org/TR/sml/.

Pedrinaci, C.; Domingue, J.: Toward the Next Wave of Services: Linked Services for the Web of Data. J. UCS 16(13): 1694-1719 (2010)

Rauf, I., Iqbal, M. Z. & Malik, Z. I. (2008) UML based Modeling of Web Service Composition - A Survey. In: Proceedings of the 6th International Conference on Software Engineering Research, Management and Applications, pp. 301-307, IEEE Press.

Roman, D., de Bruijn, J., Mocan, A., Lausen, H., Domingue, J., Bussler, C. & Fensel, D. (2006) WWW: WSMO, WSML, and WSMX in a Nutshell. In: Proceedings of the Asian Semantic Web Conference (ASWC), pp. 516-522, LNCS 4185, Springer.

Speiser, S. (2009) Semantic Usage Policies for Web Services. In: Proceedings of the 8th International Semantic Web Conference, pp. 982-989, .

Steinberg, D., Budinsky, F., Paternostro, M. & Merks, E. (2008) EMF: Eclipse Modeling Framework, 2nd Revised edition, Addison-Wesley Longman, Amsterdam. Stuhec, G. & Crawford, M. (2006) How to Solve the Business Standards Dilemma – The CCTS Standards Stack. SAP Developer Network (SDN) article, November 2006. Sun, W., Zhang, K., Chen, S., Zhang, X. & Liang, H. (2007) Software as a Service: An Integration Perspective. In: Proceedings of the Fifth International Conference on Service-Oriented Computing. LNCS 4749, Springer Berlin.

Sycara, K. (2007) Untethering Semantic Web Services. In: Semantic Web Services, Part 2. IEEE Intelligent Systems, 22(6), 8-15

Theilmann, W., Happe, J., Kotsokalis, C., Edmonds, A., Kearney, K. & Lambea, J. (2010) A Reference Architecture for Multi-Level SLA Management. Journal of Internet Engineering, to appear

Toma, I. (2010) Modeling and Ranking Semantic Web Services based on non-functional properties. PhD Thesis. Faculty of Mathematics, Computer Science and Physics of the University of Innsbruck, 2010.

Toma, I., Foxvog, D., De Paoli, F., Comerio, M., Palmonari, M. & Maurino, A. (2008) Non- functional properties in Web services. WSMO Working Draft D28.4v0.2. April 2008, http://www.wsmo.org/TR/d28/d28.4/v0.2/