Project Number:	**215219**
Project Acronym:	**SOA4All**
Project Title:	**Service Oriented Architectures for All**
Instrument:	**Integrated Project**
Thematic Priority:	**Information and Communication Technologies**

D7.6 End User Service Annotation and Context Descriptions

Activity 3:	Use Case Activities
Work Package 7:	End-user Integrated Enterprise Service Delivery Platform
Due Date:	28/02/2010
Submission Date:	
Start Date of Project:	01/03/2006
Duration of Project:	36 Months
Organization Responsible of Deliverable:	SAP
Revision:	1.0
Authors:	Juergen Vogel (SAP), Iva Tsvetkova (SAP), Sonja Meyer (SAP), Patrick Un (SAP), Tomás Pariente Lobo (ATOS), Jose Lucea (ATOS), Mateusz Radzimski (ATOS)
Reviewers:	Freddy Lecue (UNIMAN), Pierre Grenon (OU)

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	**X**

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	04.01.2010	First Version	Juergen Vogel (SAP), Iva Tsvetkova (SAP), Sonja Meyer (SAP)
0.2	27.01.2010	Updated section summaries	Patrick Un (SAP)
0.3	04.02.2010	extended related work (STASIS, SemanticGov, OntoGov); Executive Summary; revised all sections	Sonja Meyer (SAP), Juergen Vogel (SAP)
0.4	10.02.2010	Updated Section 4.2	Iva Tsvetkova (SAP)
0.5	22.02.2010	Updated Section 4.2	Iva Tsvetkova (SAP), Juergen Vogel (SAP)
0.6	22.02.2010	Input Chapter 2, 3, 4, and 5	Patrick Un (SAP)
		Input Chapter 6	Jose Lucea (ATOS), Mateusz Radzimski (ATOS), Tomas Pariente (ATOS)
0.7	23.03.2010	Revision of entire document	Juergen Vogel (SAP)
	24.03.2010	Review	Freddy Lecue (UNIMAN)
	26.03.2010	Review	Pierre Grenon (OU)
1.0	26.03.2010	Address Reviewer comments and revise all Sections	Patrick Un (SAP), Sonja Meyer (SAP), Juergen Vogel (SAP)

Table of Contents

List of Figures

List of Tables

Glossary of Acronyms

Acronym	Definition
API	Application Programming Interface
BO	Business Object
BPEL	Business Process Execution Language
BPM	Business Process Modeling
BPMN	Business Process Modeling Notation
CMS	Content Management System
CRM	Customer-Relationship Management
D	Deliverable
DSB	Distributed Service Bus
EC	European Commission
EJB	Enterprise Java Beans
EP	Enterprise Portal
ERP	Enterprise Resource Planning
ES	Enterprise Service
ESR	Enterprise Service Repository
ESB	Enterprise Service Bus
EU	European Union
EUD	End User Development
GDT	Global Data Type
GUI	Graphical User Interface
HCM	Human Capital Management
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identifier
ISO	International Organization for Standardization
IT	Information Technology
KPI	Key Performance Indicator
M	Month
NLP	Natural Language Processing
OWL	Web Ontology Language
QoS	Quality of Service

RDF	Resource Description Framework
RDFS	RDF Schema
REST	REpresentational State Transfer
SaaS	Software as a Service
SAWSDL	Semantic Annotations for WSDL
SCM	Supply Chain Management
SD	Standard Deviation
SEE	Semantic Execution Environment
SEI	Service Endpoint Interface
SME	Small and Medium Enterprise
SOA	Service-Oriented Architecture
SOA4All	Service-Oriented Architectures for All
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SRM	Supplier Relationship Management
STASIS	Software for Ambient Semantic Interoperable Services
TCO	Total Costs of Ownership
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USD	United States of America Dollars
USDL	Unified Service Description Language
WP	Work Package
WS	Web Service
WSDL	Web Services Description Language
WSML	Web Service Modeling Language
WSMO	Web Service Modeling Ontology
XI	Exchange Infrastructure
XML	Extensible Markup Language
XQuery	XML Query Language
XSPARQL	XQuery for SPARQL

Executive Summary

WP7 is one of the three SOA4All use cases and has the public sector as its target domain. It envisions an integrated service delivery platform that will be realized based on the technologies and tools developed in SOA4All. This platform will allow non-technical business users in public administrations to handle typical administrative procedures (such as a permit approval process). More specifically, using the Web-based tools of the SOA4All Studio, public servants can search, model, annotate, modify, share, analyze, and execute administrative procedures in the form of lightweight business processes. These processes may be composed of SAP Enterprise Services, public Web services (hosted by 3rd party service providers), and human activities (to be executed by end users). Thus, the main result of WP7 will be an integrated demonstrator that addresses the specific needs of public administrations. In this document, we describe the use case-specific semantic annotations required for the discovery and composition of semantic Web services with the corresponding components of SOA4All: (1) a lightweight domain ontology for SAP Enterprise Applications, (2) semantic annotations for functional and non-functional properties of SAP Enterprise Services, including (3) a novel approach for describing business semantics in a service ontology that we use to support business users in selecting and composing services meeting their specific business requirements, (4) service templates for the novel minimum service model developed in SOA4All, and (5) a working example of a contextualized business process. All artifacts described in this Deliverable (ontologies, annotated Web services, semantic queries, and processes) can be used by the technical WPs of SOA4All to validate their concepts and prototypes.

1. Introduction

WP7 is one of the three SOA4All use cases and has the public sector as its target domain. Public administrations nowadays have to deal with hundreds of different procedures (e.g., for handling a parking permit request) that are typically implemented in one or more legacy systems or are even executed manually. At the same time, the increasing number of regulatory changes and new regulations, including an increasing number of international, bilateral agreements, asks public administrations to constantly adapt their procedures in a flexible and cost-efficient way. For instance, the EU Services Directive requires administrations to implement a one stop e-Government approach where constituents can file requests for public services via a single point of contact. This single point of contact then coordinates all necessary activities, which is contrary to the current setup where the constituents themselves have the main responsibility and need to manage on their own. As a consequence, public administrations now need to adapt their service offerings to the needs of each constituent.

SOA4All investigates different key technologies (Semantic Web services, context adaptation, Web 2.0 principles) that can help to address such challenges on the basis of an advanced service-oriented architecture. WP7 envisions an open and flexible service delivery platform where administrative procedures are handled over a central Internet portal as single point of contact. Administrative procedures are interfaced by and composed of Semantic Web services. These services can be combined in different ways so that new procedures can be created or existing ones can be adapted easily. A key element for creating the content for this service delivery platform in an efficient and scalable way is the enablement of end (or business) users that resemble the large majority of employees in public administrations (and other organizations). Such business users have a detailed understanding of the procedures in their field of expertise but lack the specific programming skills required to actively consume and compose Web services. The SOA4All approach therefore is to provide end users with simple Web-based tools on top of Semantic Web technology so that they can search, model, annotate, modify, share, analyze, and execute administrative procedures on the basis of Web services. This Web 2.0 approach is also a main differentiator where SOA4All advances the current state of the art.

The main goal of WP7 is to realize this service delivery platform as an integrated demonstrator. This demonstrator is built from the components developed by the technical WPs of the project: the communication and data storage infrastructure will be provided by WP1, formalisms and tools for the semantic handling of services by WP3, services discovery and a service registry by WP5, lightweight process modeling and execution by WP6, and different Web-based end user tools by WP2. Besides the technical integration and validation, the main contribution of WP7 from a technological point of view is to investigate how so-called Enterprise Services can be integrated into the open, dynamic, lightweight, and end user-driven service platform that is envisioned by SOA4All. Such Enterprise Services, which are provided by SAP in the scope of WP7, offer complex business functionality like the management of resources or relationships with customers. But at the same time, they typically have large syntactic (i.e., WSDL-based) service interface descriptions and need to be composed in specific sequences, which makes them difficult to understand for non-expert service consumers. Thus, by investigating how to make Enterprise Services consumable for non-experts, WP7 will significantly increase the number of services that can be handled by SOA4All.

Previously, D7.2 motivated the public sector as a target domain for the SOA4All project and described how the research of SOA4All can support SAP in realizing a novel service delivery platform for the public sector by means of an exemplary use case scenario. The resulting technical and business requirements for this service delivery platform were described in

D7.1. Following, D7.3 illustrated how the SOA4All components are leveraged to build the service delivery platform and which components are developed additionally by WP7 in order to realize a complete demonstrator for the given scenario. A first prototype was delivered with D7.4.

1.1 Purpose and Structure of the Document

In this Deliverable, we describe the following use case-specific semantic annotations required for the discovery and composition of semantic Web services with the corresponding components of SOA4All[1]:

- In Chapter 2, we define a **lightweight domain ontology** for SAP Enterprise Applications. This domain ontology captures the distinct characteristics of SAP Enterprise Service (ES) and serves as a basic building block upon which more specific ontologies can be created. We present a unified approach of service annotations that is aligned with the WSMO-Lite service model and that will be reused by upper-level ontologies created in alignment with the new unified *minimal service model*; together these ontologies facilitate a novel approach to service discovery and matchmaking called *service templates*.
- In Chapter 3, we describe the **functional Web service annotations** using the WSMO-Lite service model to annotate an example SAP ES. Here we reuse the domain ontology from Chapter 2. Elements of the WSMO-Lite ontology are described with the example such as input and output messages, fault messages, service operation and service concept instance definition. Furthermore, we describe a novel approach to express pre-condition and post-condition axioms. This approach is a first step towards a lightweight service ontology to be evolved by on-going work with the minimal service model and service templates. We also discuss a hierarchy of functional classifications to assist the service discovery and matchmaking process by automatically generating of SPARQL queries to search for WSMO-Lite annotated ES.
- In Chapter 4, we extend the **Web service annotations with non-functional properties.** We introduce a **novel approach to describe business semantics**: First we introduce a dedicated service description language, the Unified Service Description Language (USDL), complete with a WSML ontology to be used in SOA4All. This ontology is then used for selecting services based on the specific, context-dependent business requirements of a SOA4All user during service invocation and service composition (process modeling). We also provide sample annotations of business semantics in USDL/WSML for selected SAP Enterprise Services (ES) and 3rd party services for the WP7 scenario.
- In Chapter 5, we discuss the **Service Templates** approach, which is currently under development in WP3, as a novel approach to simplify service discovery and matchmaking with a **simplified goal specification** focusing on the essential elements within service request or response instances in RDF. We also provide a sample service template and corresponding SPARQL query for an SAP ES.

[1] Please note that this Deliverable exceeds the internal page limit of 50 pages because we felt the necessity to provide very detailed descriptions of all semantic annotations (complete with samples and listings) together with in-depth discussions of the rationales behind in order to be understandable for a wider public audience (D7.6 is a public Deliverable).

- In Chapter 6, we present a working example of a **contextualized business process** that contains a context-sensitive goal as process activity. This goal is resolved by the WP6 process modeling and execution components to a matching Web service.
- In Annex A, we survey existing ontologies that have been or are currently defined by related European and national projects and analyze to what extent they can be reused in the scope of this use case.

All artifacts described in this Deliverable (ontologies, annotated Web services, semantic queries, and processes) can be used by the technical WPs of SOA4All to validate their concepts and prototypes.

1.2 Future Work

The final version of the demonstrator will be delivered with D7.5 in (M33). This demonstrator will leverage the semantics described in this deliverable and support the business users in achieving their goals with our service delivery platform. Following, the last three months of the project will focus on the concluding technical validation and the user experience evaluation of the achieved results. The corresponding report D7.7 will be available in February 2011 (M36).

2. Domain Ontologies for the Use Case

Generic domain independent ontologies serve as basic building blocks upon which more detailed domain-specific ontologies can be built. In the following sections, we present a unified approach of service annotations that is aligned with the WSMO-Lite service model[2] for the WP7 use case scenario[3]Furthermore, the fundamental service ontologies in WSMO-Lite service model will be reused by upper-level ontologies in alignment with the unified minimal service model [Lambert2010]; together they facilitate a novel approach to service discovery and matchmaking called *service templates*.

We are primarily interested in the distinct characteristic of SAP Enterprise Service (ES) (e.g., for accessing the functionality of an SAP ERP system):

- Rich information model.
- Process orientation of service consumption.
- Enterprise middleware-based infrastructure of service delivery.
- Process-coupled service behavioral model.
- Complex service interface definition.
- Complex data parameter and types based on so-called Business Objects (BO) with implicit data value and range information.
- Structured information incorporating data semantics and type information for each individual BO in ES.
- Intra-dependencies between data parameter types within an individual service message.
- Inter-dependencies between data among ES, e.g., certain data parameters in a request require values that must be obtained implicitly from data value in response of previously invoked service.
- Stateful information is required by ES due to process orientation. Certain stateful parameters such as change state identifier is necessary.

These aspects of ES influence not only the behavior of the ES but also have consequences for the design and implementation of the corresponding information model and business data objects which the ES manipulate. Before designing an ontology that suitably reflects this enterprise information model, we briefly describe the information model of SAP enterprise applications. Subsequently, we design an ontology with regard to this information model.

2.1 Enterprise Application Information Model

SAP ES are developed in a way that is aligned tightly with the underlying process business logics, inheriting the general notion of ES technology infrastructure of the SAP enterprise application platform. Standard technology and protocol for service interface description such as WSDL, HTTP, synchronous or asynchronous invocation, different message exchange

[2] See SOA4All Deliverables D3.4.2 WSMO-Lite: Lightweight Service Descriptions for the Web and D1.2.1 WSMO Grounding in SAWSDL.

[3] See SOA4All Deliverables D7.2 Scenario Definition and D7.3 End User Service Design.

patterns and service registry technology such as UDDI are used.

Currently the WS-_ stack of extended Web services specifications and standards such as WS-BusinessActivity, WS-CDL, WS-Coordination, etc. are supported by the ES middleware, as well as support of WS-BPEL and a set of important Web services transaction standards have been built into current release of the SAP NetWeaver Application Suite. What characterizes these SAP ES is that they are organized into functional bundles according to the business designation and domain of interest such as financial ledger (FI), customer relationship management (CRM), human resource (HR), etc. The aim of enterprise process and information modeling is to specify a meta-model for business process workflow and dataflow specifications by using a systematic approach to structure, visualize and document the relationships, dependencies and interactions between components of an enterprise system.

In order to minimize redundancy in the different levels of enterprise information model and facilitate reuse of common components, it is necessary to structure enterprise data more transparently and modularize them into a set of manageable units. The fundamental idea is to use a clear abstraction to model data that is independent from specific implementation details. The data of process, workflow and control flow is modeled using Business Objects (BO) in SAP systems [Snabe2009]. BO describe the data of a process as a fundamental entity. They distinguish between the abstract data and the implementation of business logic to manipulate that data and are structured in such a way to enhance good modeling practice of the business process, avoiding duplication of data model or overlapping definitions of information scope. BO represent specific views of data of a well-defined business domain and are the cornerstones of the principle of data modeling used by SAP enterprise systems. A BO can be combined to form more complex data structures. The lifespan of a BO is another aspect to characterize in data modeling with the difference between master transactional data called master data transactional BO; and transient data called process objects. BO that work with essentially persistent data over the entire system lifespan over a long period of time are called master data transactional BO.

ES need to work with and manipulate data represented by BO. Therefore, the informational modeling of dataflow ES are intrinsically linked with the modeling of BO; both transient process objects and master data BO. The modeling process with BO is carried out on the ARIS modeling platform products[4]. In an ARIS model an abstraction process is used to specify components which work with data of a business domain. Separation in structural modeling and semantic modeling of data is crucial in order to distinguish data type, data value with the business semantic of these data. Identifying self-contained parts of a business value chain using this modeling approach is relevant and crucial in subsequent implementation of cross organizational business-to-business processes [Snabe2009].

A process component in SAP system is defined as the part of a value chain that performs for a specific task in a business area in an enterprise and it acts as a unit to group BO logically. Each BO belongs to one process component. Additionally the level of granularity of information modeling using process components is also relevant because the type or structure of BO is closely connected with the modeling granularity chosen and it is important to ensure that there is no overlapped part of process components containing duplicated information by facilitating good structural organization and information reuse by allowing shared data from different process components to become accessible.

Accessing data in a process component implies accessing information of relevant BO of the

[4] See http://www.ids-scheer.com/en/ARIS_ARIS_Platform/3730.html

component. There are two types of access patterns: asynchronous access to process component is used for inter-process communications between process components of different business areas and value chains. Synchronous access pattern depends on accessing interrelated process components of similar business areas of different value chains. Access pattern of process components is modeled in the ARIS platform as service interfaces and operations that mimic access of BO. Operations on BO are assigned to the interface of BO. A service interface is used to group operations in order to expose BO data via the access operations to external components.

While the process component model is used to describe the inner workings of a business process component by specifying data model through BO as well as the corresponding service interfaces to access them, a deployment unit is used to group all process components that belong to a specific business scenario so that they can be installed together and configured for a certain process enactment. In the process component integration and deployment model, deployment units are specified and are mapped to the corresponding process components via the ARIS platform tools.

2.2 Core and Global Data Types

BO consist of a distinct semantic structure with specification of separation of data typing, access operations and manipulation logic to model business process components efficiently. Implementation details of the operations defined in the service interfaces are independent of the specific typing scope and reflects the semantic information model of business data with distinct characteristics and behavior. On the other hand, a conceptual data model has been introduced to categorize and structure data types according to the UN/CEFACT Core Component Technical Specification Standard (CCTS/ISO 15000-5)[5]. Data types which conform to the standard are called Core Data Types (CDT) across SAP applications and they implement the separation of data typing information and corresponding business semantics in a unified manner. Each CDT defines exactly one primary component known as the content component across SAP applications.

CDT are syntax neutral and are capable to represent atomic, from the most generic to the most specific pieces of process information within a business domain. They are generalized data types based on primitive data types such as the well-known primitive types in W3C XML Schema. CDT are different than the W3C primitive types because they carry attributes defining type properties in the data source domain and value range information defining the valid and concrete value within a certain application domain. Each CDT can be identified with an application wide unique representation term which is not only applicable to SAP applications but also in other domains. CDT serve as the basic building blocks to form more elaborated complex types known as Global Data Types (GDT).

GDT are divided into basic and aggregated types. While the basic type is built directly on other CDT, the aggregated data type contains list of CDT or aggregation of elements of another GDT, i.e., it can be seen as a composite type. GDT contain business semantics that is used across a wide range of SAP applications. Figure 1 shows the aforementioned data types in relation with business components, BO and service components such as message types and message elements of an enterprise application. It is obvious that both the BO and ES depend on the notion of CDT and GDT.

[5] See www.unece.org/**cefact**/ebxml/CCTS_V2-01_Final.pdf

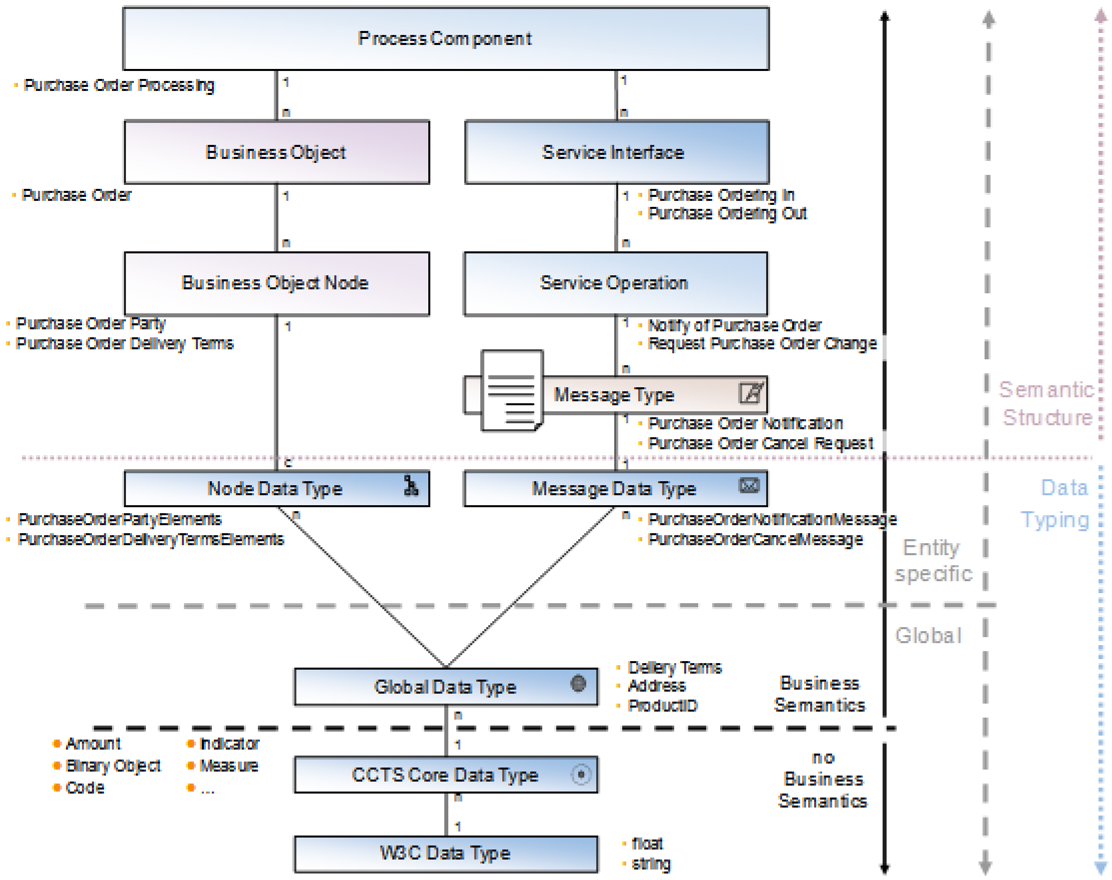

Figure 1: Semantic Structure and Data Typing of ES and Business Components

Basic business semantics are contained within the scope of the semantic structure of the information model with a BO centric organization. The node data types and message data types of process components are built on top of the GDT. Inheriting the incurred business semantics of them, node data types and message data types possess business semantics that is aligned with that of the underlying layer. Since BO of process component also deal with specific business context, upper level context-specific data types are derived directly from the GDT.

Common business semantics of GDT are modeled with semantic templates in ARIS model which specifies all the elements and attributes of a GDT without redundancy where attributes define elementary features of the underlying data types. Templates have no implementation and it ensures the consistency of semantics of the described GDT. Semantic templates specify value range with constraints on valid content. They inherit the data structure and integrity conditions from their constituents. The figure above illustrates example of BO representing a range of ES messages using concepts such as purchase order related BO on one side and the corresponding service message types on the other side. It explains the relationships of BO to service data types in service messages from a functional and operational perspective.

2.3 Enterprise Service Behavioral Model

The ES behavioral model can be characterized with a domain specific service ontology that

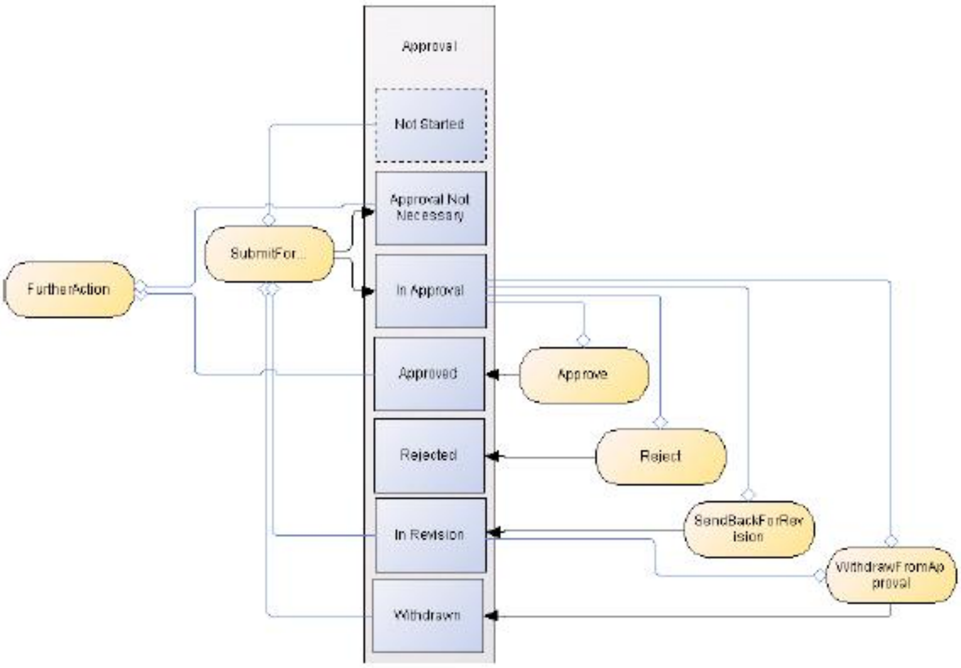

Figure 2: SAP Status and Action Management Model

describes required GDT, BO, node types, message types, input and output messages semantically. Moreover the *conditions* of service invocation are constituent part of the service behavioral model, for instance, what a-priori condition and constraint must hold before an ES is invoked and what effects will the invocation of the ES generate in the application domain. Furthermore, a-posteriori constraints after service invocation must also be characterized correctly. This stateful behavior of ES is characterized by the *status action management model* (SAMM) in SAP applications. The previously mentioned enterprise information model and structured ontology of the data types in the application domain is not yet sufficient when it comes to describe the service behavioral semantics.

The behavioral aspects of ES conform to the stateful SAMM and must be described semantically which allow a formal description of the constraints. Figure 2 illustrates an excerpt of SAMM for constraint descriptions of an example Approval business activity. It specifies the constraints associated with each individual operation with regard to the BO and data types which this activity manipulates. In order to enforce the invocation constraints, *guards* are attached to the operations to check if the associated constraints hold before invocation and that all specified effects have appeared afterward.

From the process perspective, process control flow characterizes the behavior of underlying resolvable ES at runtime. Process behavioral model defines the constraints of the control flow rather than directly of each individual service; with the intention that no redundant information is defined for both process and service and considering that ES are extended interface of core functionality of enterprise applications but not just simply individual units of their own. This fact has implication for the behavioral model of ES so that we view these services as black boxes with predefined set of a-priori and a-posteriori constraints that are prescribed by the process model. At runtime, control flow influences the creation of activity instances based on evaluation of these constraints and thus the binding and invocation of appropriate ES.

If we adopt an abstract view of behavioral model of ES from a pure functional perspective and view such services as black boxes, we can define the constraints for them with basically four types of information:

- constraint on input message type and value domain,
- constraint on output message type and value domain,
- constraint on the pre-condition of invocation, expressed using the vocabularies that are available for describing the BO and message components of input messages, thus linking behavioral description in this regard to the information model of the ES,
- constraint on the post-condition and effects after invocation, also expressed using the known set of vocabularies of the ES information model.

If we view the result of a process step at runtime as a contribution of information into the *information space* of an individual ES, the information must comply with the ES information model of the application domain. It either enriches the information space by introducing new BO instances into the process instance or changes it by altering existing BO instances. In this regard, it is therefore correct to link the behavioral description of ES to the context of the service message exchange and the type constraints on constituent parts of these messages. The effect of an invocation can therefore be characterized correctly using the known BO instances and incurred data type instances in the information space. This fact will influence our subsequent modeling of semantics with ontological concepts of the information space using RDF/S and the WSMO-Lite lightweight semantic service model.

2.4 Requirements for a Semantic Conceptualization of ES

In the sections above, we have described the principles of information modeling of ES in terms of separation of semantic structure and data typing information of process components, BO and the individual data types. In this section, we discuss how to express the information model semantically so that we can effectively reflect the essential elements of the existing enterprise information model to align it with the lightweight semantic service model. Considering the fact that the entire business information model of ES possesses an opulently rich information structure, there is a tradeoff when designing the conceptualization and deriving concepts from existing rich model, i.e., the tradeoff between scalability of the derived model and expressivity of the formalism used to model the information.

Since SOA4All will adopt RDF/S as semantic language and WSMO-Lite based service annotations and data grounding, which to a certain extend has limitations in semantic expressiveness for instance compared to more expressive language such as OWL. However for better scalability in service discovery and matchmaking, for instance, for the very large number of services targeted by SOA4All. With high expressiveness of formalism can incur intractability and suboptimal reasoning performance [Sattler2005], it is of high priority to trade simpler expressiveness, tractability, decidability and scalability against higher semantic expressiveness, non-deterministism or undecidability in reasoning[6]. Hence we have to extract the concepts from the existing enterprise business information model with the expressivity of RDF in mind in order to harmonize our conceptualization with the available expressiveness of this formalism, yet retaining tractability and scalability in reasoning. In this course it is necessary to structure information with a logical layering and modularization approach for effective reuse and to avoid information redundancy.

[6] Such as for OWL-Full.

There are two general requirements: (1) to structure the core business concepts such as process model, process components, process constraints within an upper level fundament ontology which can be reused; (2) to extract the relevant concepts for the functional aspects of describing the operational requirements of ES such as message elements, BO, BO nodes as well as typing and type mapping information from the existing business information model and to express them in an RDF/S commonly shared ontology. This follows the rationale of lightweight approach of SOA4All by adopting a pragmatic bottom-up approach in the

Figure 3: A Fundamental Business Domain Concept Hierarchy (Partial)

functional description of ES by selecting all relevant concepts required with ES operational considerations. Core process and ES concepts are organized in the upper level ontology `sap_esi_layer_business_domain_fundamental_ontology.n3`. It uses RDF/S and is serialized in the W3C recommended Notation 3/Turtle syntax. The fundamental ontology is structured with the most generic business domain concepts reside in the root concept, from which more specific concepts and terminology of the enterprise business domain are derived. Figure 3 shows part of a visualization of the conceptual hierarchy[7].

From a reuse point of view, an application domain specific ontology can derive more specific concepts from this fundamental ontology. Moreover, it follows the clean separation of the definition of business semantics of concepts from the data typing aspects of these business entities as described . While the concepts in the ontology such as `BusinessDomainConcept`, `BusinessObject` and `BusinessObjectNode`, etc. conceptually represents the existing entities in the core business information model, the main advantage of adopting this principle is to ensure that the existing semantics in the business information model is uphold and preserved and consequently reflected accordingly in the fundamental business domain concept hierarchy.

[7] Due to brevity reason, the conceptual hierarchy is not shown with RDF properties.

Data types as described previously by the set of W3C core data types, CCTS-conforming data type specification, the SAP CDT and GDT are represented with corresponding conceptualizations in the fundamental ontology. Not only the business domain concepts but also relations are captured wrt correct semantics conforming to the existing core model.

2.5 Lightweight Semantic Enterprise Service Ontology

Semantic ES information modeling needs further refined knowledge of the business data and typing information. The semantic modeling process takes into account the WSMO-Lite service annotation and grounding mechanisms, i.e., by providing a required set of RDF concept instances that are created from the required set of ontology concepts describing the business data (in the input and output messages) exchanged in ES. Thereby WSMO-Lite annotation uses the SAWSDL reference mechanism to semantically link the defined business data concepts to the XML Schema types of the ES in their WSDL interface definitions. At runtime, in addition to using concepts from our fundamental ontology, additional application specific business data types and BO concepts, are captured in a middle level, modularized domain specific ontology shared among ES instances. These concepts are instantiated by the semantic execution environment (SEE) of SOA4All and are available as RDF instances in the Semantic Space. Because of the close resemblance of the business data and BO concepts with the actual XML types, we have decided to closely follow the naming scheme of these entities and name them accordingly in the middle level domain specific shared ontology called:

`sap_esi_layer_business_domain_specific_shared_ontology.n3`.

Together with this Deliverable, we have provided a common set of shared and non-redundant conceptualizations of all required business entities and business data types in this ontology hierarchy for the WP7 use case scenario; with roughly 570 RDF/S classes for business entity concepts, approximately 390 RDF/S object properties and roughly 690 individual instances. In the our process a pragmatic bottom-up approach is followed by first examining the ES interface definitions and then extracting the minimal set of required and shared entities in order to capture them in corresponding semantic entities. Later refactoring is required to identify and avoid redundancy and structure them in modularized units for better reuse. In this layering approach, the principle of separation of business semantic from data typing information is extensively taken care of. Both the business concepts and relations serve to extend the fundamental ontology in such a way that business entities extend the GDT domain concepts to represent typing coherence with the W3C and CCTS-conformity. SAP specific business data types as shown in an excerpt of the ontology visualization in Figure 5. It shows that data typing information of the business entities are conceptually related to the core `GlobalDataType` concept described previously. The modeling of associated business semantics of these business entities is captured in the derivation of another concept hierarchy of these types simultaneously from the `BusinessObjectNode` concept as shown in Figure 4.

Individual WSMO-Lite service ontology can reuse concepts in the upper ontology by creating instances of concepts and relations at runtime. Provided that the type mapping and grounding information is created between XML types and corresponding semantic concepts at design time, the SEE of SOA4All is capable of providing the necessary grounding implementation and proper type mapping to enforce necessary data type and semantic concepts transformation by using the transformation services of WP2[8]. Transformation service is referenced using the SAWSDL 'liftingSchemaMapping' and

[8] See SOA4All Deliverable of WP2: D2.2.1 and D2.2.2 on service consumption platform.

'loweringSchemaMapping' mechanism to refer transformation to dedicated SOA4All transformation services. It is necessary to device the conceptualization and provide a rich set of corresponding semantic concepts and relations to facilitate semantic service operation at runtime.

Figure 5: Concept hierarchy of operational aspect of business types deriving from GlobalDataType concept of the upper-level fundamental ontology.

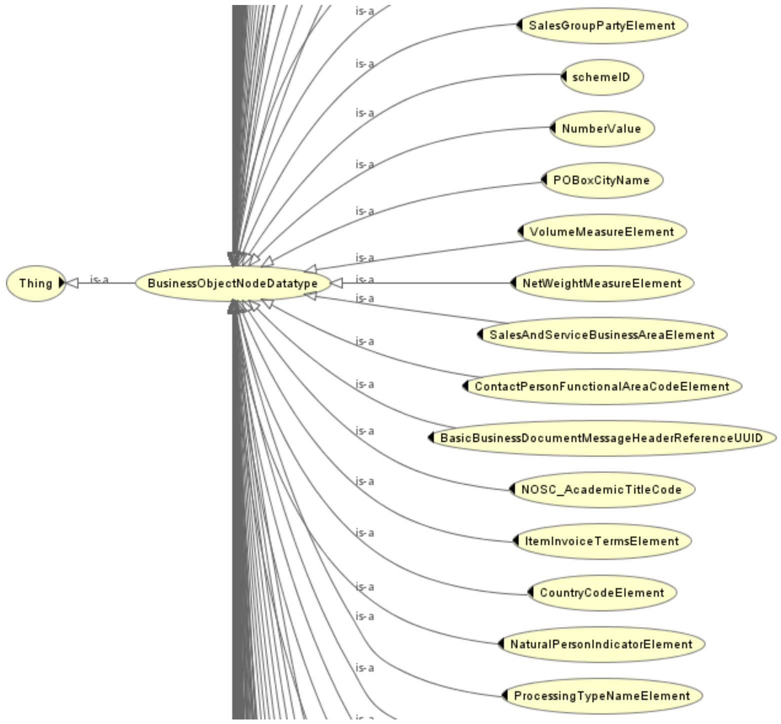

Figure 4: Concept hierarchy of BO nodes deriving from the BusinessObjectNode concept of the upper ontology.

We have referred to our structure of conceptualization for our business information model as a layering and modularized approach since the shared ontology reuses and extend the fundamental ontology. Furthermore the shared ontology serves in the middle layer as necessary stratum of conceptualizations for the bottom layer application specific WSMO-Lite ES ontologies. The conceptual hierarchy of this setup is shown in Figure 6.

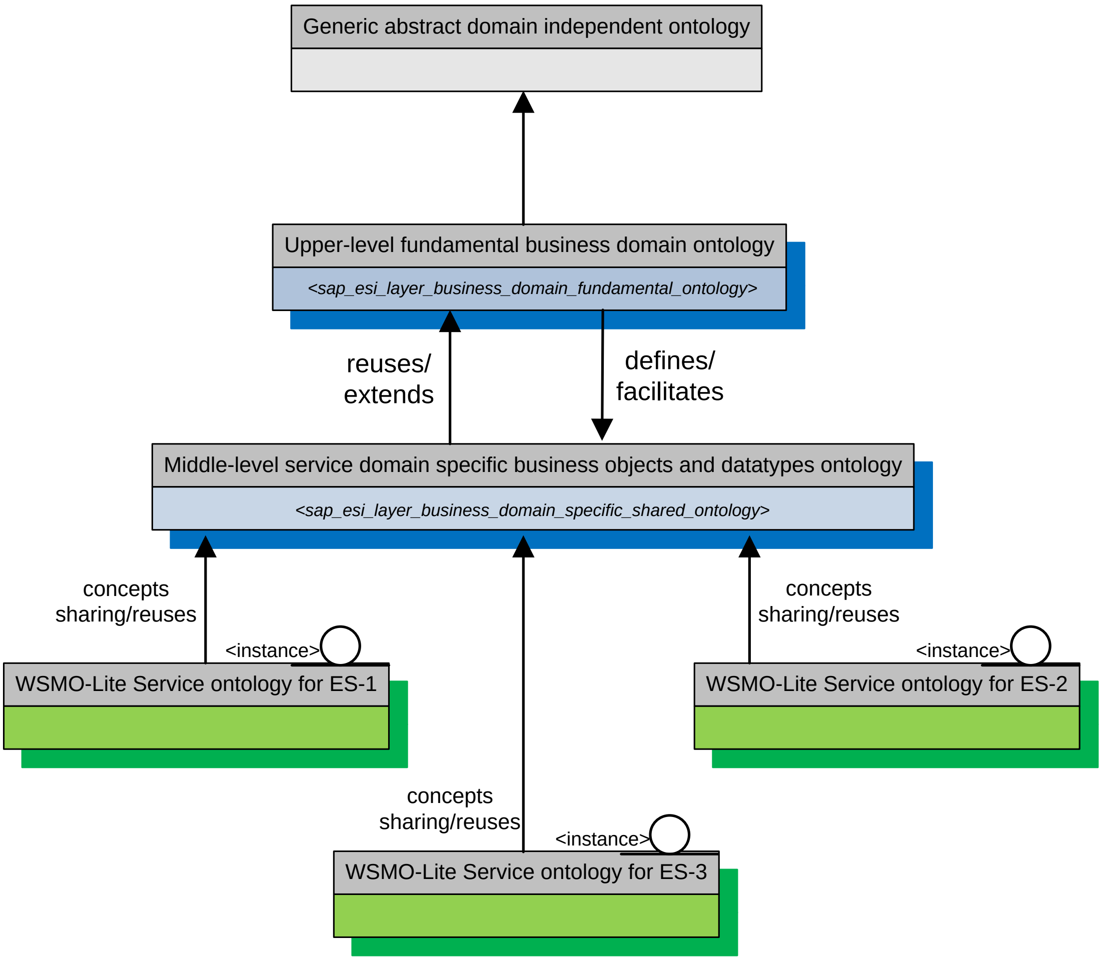

Figure 6: Conceptual Ontology Hierarchy: From Generic to Domain-specific ES Ontologies.

3. Service Annotation with Functional Properties

This chapter describes the functional service annotations using the WSMO-Lite service model to annotate an example SAP ES within the use case scenario. Thereby emphasis is placed on the reuse of the previously discussed domain specific shared ontology. Elements of the WSMO-Lite [Kopecky2008] ontology are described with the example such as input and output messages, fault messages, service operation and service concept instance definition. Furthermore, we describe service behavior and express pre-condition and post-condition axioms similar to those of the WSMO service ontology using the WSMO-Lite `AxiomLiteral` construct. Though this approach is merely a first step to be subsequently replaced by a work-in-progress more lightweight service model called minimal service model and service templates [Lambert2010]. We also describe a hierarchy of functional classifications to assist the service discovery and matchmaking process. Finally, we have devised a set of non-functional parameters to further assist service selection based on both functional and non-functional information for the ES.

In Section 3.4, we will focus on the new minimal service model to consolidate service discovery by employing the work-in-progress new concept of service templates to wrap a service functional description in a way which allows automatic generation of concrete or parameterized SPARQL queries to search for WSMO-Lite annotated ES which also conform to the minimal service model. This new approach has the advantage that SPARQL queries are suitable for querying RDF/S data efficiently and provide a more lightweight service discovery and matchmaking mechanism than the goal based stateful model hitherto known in the WSMO [Roman2005] family of service ontology, which employs condition-based service composition in addition to service matchmaking.

3.1 Service Interface

We discuss the annotation of functional aspects with an example ES:

`sap_esi_layer_es_address_basic_data_by_name_and_address.wsdl`

Its WSDL interface is found in the SOA4All project repository in the sub-folder structures

http://svn.sti2.at/soa4all/trunk/soa4all-usecases/WP7/ontology/D7.6/SAWSDL.

It is used in the use case scenario to find personal information that is stored in an SAP ERP system by using either a name or a known address as search criteria. Functionally it is classified as related to CRM functionality. It contains a single operation with a pair of message elements and type definitions. The XML types therein represent a very rich set of enterprise business data types with rich interface definition of both mandatory and optional parameters allowed in the message exchange. Complex interface with rich data types are common characteristics found in a wide range of other SAP ES. Throughout this and the next section, functional annotations will be discussed based on this example ES.

The service element is defined as the following:

```
<wsdl:service name="ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService"
    sawsdl:modelReference="urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:
    sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite#
    ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService">
     <wsdl:port name="ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePortType"
binding="tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseSOAPBinding">
     <soap:address
location="http://localhost/ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService" />
   </wsdl:port>
</wsdl:service>
```

The ES defines a single operation which accepts input request message and generates output message of the XML type as shown in highlighted text in the following listing. There is also an outbound fault message defined in case of error or processing exception.

```
<wsdl:portType name="ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePortType">
  <wsdl:operation name="ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation">
   <sawsdl:attrExtensions sawsdl:modelReference="
    urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:
    sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite#
    ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation" />
     <wsdl:input message="tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync" />
     <wsdl:output message="tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync" />
     <wsdl:fault name="exception00" message="tns:exception00" />
  </wsdl:operation>
</wsdl:portType>
```

There are SAWSDL references both for the service element and operation, i.e. we define in the corresponding WSMO-Lite service ontology the corresponding instances which are instances of the `wsl:Service` and `wsl:Operation` classes and use the SAWSDL reference mechanism to link in these instances in WSDL.

```
@prefix tns:<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
tns:WSMOlite_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceOntology  a
wsl:Ontology;
    rdfs:comment "WSMO-Lite (CustomerERPAddressBasicDataByNameAndAddressQueryResponse_InService)";
    rdfs:isDefinedBy
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:sap:domain#sharedDomainSpecificOntology>;
    rdfs:seeAlso
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:domain#sharedDomainSpecificOntology>;
    rdfs:isDefinedBy
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:domain#commonFundamentalOntology>;
    rdfs:seeAlso
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:domain#commonFundamentalOntology>.
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation a wsl:Operation;
 sawsdl:modelReference <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:
```

```
sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite#
ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation>;
    wsl:hasInputMessage tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync;
    wsl:hasOutputMessage tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync;
    wsl:hasOutputFault tns:exception00.
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService a wsl:Service;
    wsl:hasOperation tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation;
    wsl:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#sharedDomainSpecificOntology>;
    wsl:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#commonFundamentalOntology>;
    wsl:usesOntology
tns:WSMOlite_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceOntology;
    sawsdl:modelReference <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:
sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite#
ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService>;
    rdfs:isDefinedBy
    <http://localhost/ ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService?wsdl>.
```

It is worth mentioning that the semantic operational model is defined with three slots associating input message concept instance, output message concept instance and the fault message concept instance via RDF properties in the WSMO-Lite service ontology. The operation concept instance itself is linked to the service concept instance subsequently via the `wsl:hasOperation` property. Related ontology, especially the middle-level ontology and the upper level ontology can be declaratively linked in within a WSMO-Lite service ontology using the `wsl:usesOntology` property.

The `rdfs:isDefinedBy` property takes in its range an instance of `rdfs:Resource` which is used here to assert an instance of the service URL. This information is handy for lightweight service discovery agent which can use SPARQL to retrieve location and invoke the ES easily.

We must specify the messages according to the WSDL interface definition shown in the following listing.

```
<wsdl:message name="CustomerERPAddressBasicDataByNameAndAddressQuery_sync">
   <wsdl:part name="parameters" element="tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync"
   sawsdl:modelReference="urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:
sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite#CustomerERPAddressBasicDataByNameAndAddressQuery_syncElement
</wsdl:message>
<wsdl:message name="CustomerERPAddressBasicDataByNameAndAddressResponse_sync">
   <wsdl:part name="parameters" element="tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync"
   sawsdl:modelReference="urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:
sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite#CustomerERPAddressBasicDataByNameAndAddressResponse_syncElem
/>
</wsdl:message>
<wsdl:message name="exception00">
   <wsdl:part  name="exception00" element="n0:StandardMessageFault"
   sawsdl:modelReference="urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:
sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite#exception00Element" />
```

```
</wsdl:message>
```

In the example we must also define concept instances for the input message, output message and fault message as shown in WSDL, bearing in mind that .each message name as highlighted in the listing represents an identifier for a message wrapper element that is carried in the payload of a SOAP message. Take an input message for instance, this request body contains the wrapper element according to specific process style such as document style

`CustomerERPAddressBasicDataByNameAndAddressQuery_syncElement`

This message wrapper element consists of a complex set of types which for brevity reason is omitted in this document. The XML Schema definition of these complex types conforms to the defined structure and each SOAP request payload should contain these XML type instances as well. In the semantic space, our middle-level domain specific shared ontology defines all the required concepts that maps to these complex types. It is up to our decision whether we choose to reflect the XML data type structure as it is in the corresponding semantic concepts. We have found that information structure prescribed by XML Schema types and RDF/S concepts are not similar and therefore one can reflect the Schema type structure in RDF, for instance, by defining internal structure using RDF properties within one concept class definition in order to link in sub-concepts which correspond to the XML sub-elements or attributes of an XML `complexType`. Though it is rather tedious task to define all these concepts as described previously, we have found that since these concepts are required at runtime by the Semantic Space and SOA4All Reasoner, it is necessary to define them exhaustively once and reuse them across a wide range of ES ontologies. An example of the `complexType` *Log* is defined in XML Schema as:

```
<xsd:complexType name="Log"
   <xsd:sequence>
      <xsd:element name="BusinessDocumentProcessingResultCode" type="ProcessingResultCode" minOccurs="0"/>
      <xsd:element name="MaximumLogItemSeverityCode" type="LogItemSeverityCode" minOccurs="0"/>
      <xsd:element name="Item" type="LogItem" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Here we have declared these corresponding semantic concepts in the shared ontology which is shown in the following excerpt listing:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix sapgdt: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:globaldatatypes#>.
@prefix sapbondt: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:bonodedatatype#>.
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#>.

tns:Log a rdfs:Class;
    rdfs:subClassOf   ccts:ISO15000CCTSConcept;
    rdfs:subClassOf   sapgdt:GlobalDatatype;
```

```
    rdfs:subClassOf   sapbondt:BusinessObjectNodeDatatype;
    tns:hasBusinessDocumentProcessingResultCode   tns:BusinessDocumentProcessingResultCode;
    tns:hasMaximumLogItemSeverityCode   tns:MaximumLogItemSeverityCode;
    tns:hasItem   tns:Item.
```

By using RDF properties to facilitate the definition of internal structure, represented here as information slots, linking the other three semantic concepts corresponding to the internal elements of the XML `complexType`. For brevity we do not show the further concept declarations in the shared ontology and users can find them in the project repository.

Subsequently we show the input/output messages and fault message in the following WSMO-Lite service ontology listing of our example ES. Service messages are instances of the WSMO-Lite `wsl:Message` class.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#>.

tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync a wsl:Message;
    sawsdl:modelReference tns:CustomerERPAddressBasicDataByNameAndAddressQuery_syncElement;
    sawsdl:liftingSchemaMapping
<urn:un:eu:soa4all:transform:interoperation:standard:soa4all_wp2_transformation_service#semanticTransformationService>.

tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync a wsl:Message;
    sawsdl:modelReference tns:CustomerERPAddressBasicDataByNameAndAddressResponse_syncElement;
    sawsdl:loweringSchemaMapping
<urn:un:eu:soa4all:transform:interoperation:standard:soa4all_wp2_transformation_service#semanticTransformationService>.

tns:exception00 a wsl:Message;
    sawsdl:modelReference tns:exception00Element;
    sawsdl:loweringSchemaMapping
<urn:un:eu:soa4all:transform:interoperation:standard:soa4all_wp2_transformation_service#semanticTransformationService>.
```

At runtime, the SOA4All reasoning service obtains information about the location of the schema lifting and lowering facilities which are part of the data grounding facilities provided in the form of platform services developed in WP2. These data grounding facilities serve as transformation components and are capable of *lifting* operation, i.e. converting XML instances in a request to the corresponding semantic concept instances and *lowering* operation on service messages, i.e. transforming semantic concept instances to their corresponding XML instances in response and outbound fault messages.

3.2 Pre- and Postconditions

In the WSMO semantic service model [Roman2005], preconditions and postconditions are key parts of the service discovery, service matchmaking and service composition mechanism by utilizing logical axioms to encode necessary constraints into conditions literals that are interpreted by logic reasoners. Logical constraints are contained in an entity called WSMO goal to express a-priori and a-posteriori constraints on invocation with regard to the desire of a service consumer. Similarly precondition and postcondition axioms are also provided in the WSMO ontology instance of a WSMO service annotation to express service capability and behavior. Matching goal constraints with the service constraints and capability is the central notion of the WSMO service behavioral model. It serves to enable a goal driven approach in service matchmaking and service composition. The drawback of this approach is however the intensive expertise required in the axiomatization of the domain model and constraint creation process in which a domain expert is usually required to write down valid and sound logical axioms correctly for an application domain; this task is no job for a layman. This disadvantage has negative impact on the scalability and availability of WSMO based semantic service consumption and provision [Kopecky2008].

In an attempt to create a lightweight service model WSMO-Lite strives to achieve the balance of lightweight service interface functional and constraint description by alleviating laymen to write down the logical axioms by themselves; instead the WSMO-Lite service model provides a lighter service discovery and matchmaking model using RDF/S. It specifies two classes to state precondition and postcondition in the service interface: `wsl:Condition` and `wsl:Effect`. A WSMO-Lite service ontology instance should contain an instance of `wsl:Condition` to declare a precondition constraint and an instance of `wsl:Effect` to declare a postcondition constraint or effect (In WSMO-Lite postcondition and effect is viewed as similar entities). These classes are defined in WSMO-Lite as follows:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#> .
wsl:Condition a rdfs:Class .
wsl:Effect a rdfs:Class .
```

In [Kopecky2008] it is shown that a WSMO-Lite service ontology can but is not mandatorily required to provide instances of `wsl:Condition` and `wsl:Effect` in the `wsl:Service` instance. This rather loose constraint on the usage of `wsl:Condition` and `wsl:Effect` provides a certain degree of flexibility for WSMO-Lite based service annotation. Furthermore, both type of these RDF/S classes use an extra information slot via an RDF property `rdf:value` in their definition to link in a string literal of type `rdfs:Liter`al or the WSML-oriented sub-concept `wsml:AxiomLiteral` in order to provide possibility to write down an arbitrary syntax of logical axioms which may be interpreted by some external reasoners. The WSMO-Lite service model is agnostic about the syntax or type of this type of logical axioms as shown in the following definition listing.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#> .
@prefix wsml: <http://www.wsmo.org/wsml/wsml_syntax#> .
rdf:value a rdf:Property ;
    rdfs:domain rdfs:Resource ;
    rdfs:range rdfs:Resource
wsml:AxiomLiteral rdfs:subClassOf rdfs:Literal
wsl:Condition a rdfs:Class ;
    rdf:value rdfs:Literal ;
    rdf:value wsml:AxiomLiteral .
```

```
wsl:Effect a rdfs:Class ;
    rdf:value rdfs:Literal ;
    rdf:value wsml:AxiomLiteral .
```

In fact the flexibility in defining condition constraints and the nature of syntax agnostic notation in WSMO-Lite service model pushes the logical interpretation of them outside the service model, thus making the model more lightweight by focusing on the essential service annotation artifacts.

On the other hand, if we write down the conditions and effect in our example WSMO-Lite service ontology it is still required to have a certain degree of expertise to extract these constraints from the ES behavior. There are two essential approaches:

- Conceive of a holistic logical axiomatization of the complete process scenario and derive the constraints as preconditions and postconditions for each individual service from the axiomatization; subsequently use a logical formalism, for instance, using non-monotonic NAF[9] logic programming (LP) rules to express each constraint as a rule constituting of a series of conjunctive logical atoms. Certainly this approach does not seem pragmatic regarding the high expertise demand on laymen to formulate the axiomatization or the rules correctly.
- Adopt a bottom-up approach by viewing each individual ES as a black box and derive the precondition and postcondition/effect from its behavior and describe them in terms of the know entities of the service, i.e. in terms of the message instances and create a link to the change of the state of the world by describing these possible changes using known business data concepts from our business domain shared ontology. Business data concept instances in the request message instance, which are required before the invocation of an ES, serve to describe precondition and similarly business data concept instances in the response message instance describe the effect after invocation of the ES. This approach is a derivative from the object-based concept declaration and tries to express an axiom by using a conjunctive or disjunctive series of concept instance in the axioms; thereby checking the concept instances and it value structure in predicates. This latter approach is pragmatic in the sense that simpler reasoning process such as query by SPARQL against a known knowledge base (set of known ES ontology instances with corresponding business data instances in the Semantic Space) can be used for lightweight service discovery and matchmaking.

We have adopted the latter approach and an excerpt of the WSMO-Lite service ontology is shown in the following listing with the precondition instance and effect instance.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#> .
@prefix wsml: <http://www.wsmo.org/wsml/wsml_syntax#> .
@prefix saperpsharedonto: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#>
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServicePrecondition a wsl:Condition;
    rdf:value """
        (
```

9 Abbreviation for negation-as-failure.

```
          ?Name[saperpsharedonto:FirstLineName      wsml:hasValue      ?firstLineName]      wsml:memberOf
saperpsharedonto:Name or
          ?Name[saperpsharedonto:SecondLineName      wsml:hasValue      ?secondLineName]      wsml:memberOf
saperpsharedonto:Name
         )
         or
         (
          ?PhysicalAddress[saperpsharedonto:CountryCode    wsml:hasValue    ?countryCode]    wsml:memberOf
saperpsharedonto:PhysicalAddress or
          ?PhysicalAddress[saperpsharedonto:RegionCode    wsml:hasValue    ?regionCode]    wsml:memberOf
saperpsharedonto:PhysicalAddress or
          ?PhysicalAddress[saperpsharedonto:StreetPostalCode       wsml:hasValue       ?streetPostalCode]
wsml:memberOf saperpsharedonto:PhysicalAddress or
          ?PhysicalAddress[saperpsharedonto:CityName      wsml:hasValue      ?cityName]      wsml:memberOf
saperpsharedonto:PhysicalAddress or
          ?PhysicalAddress[saperpsharedonto:DistrictName   wsml:hasValue   ?districtName]   wsml:memberOf
saperpsharedonto:PhysicalAddress or
          ?PhysicalAddress[saperpsharedonto:POBoxID      wsml:hasValue      ?pOBoxID]      wsml:memberOf
saperpsharedonto:PhysicalAddress or
          ?PhysicalAddress[saperpsharedonto:StreetName    wsml:hasValue    ?streetName]    wsml:memberOf
saperpsharedonto:PhysicalAddress or
          ?PhysicalAddress[saperpsharedonto:HouseID      wsml:hasValue      ?houseID]      wsml:memberOf
saperpsharedonto:PhysicalAddress
         )
    """^^wsml:AxiomLiteral.
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceEffect a wsl:Effect;
 rdf:value """
         (
          ?CustomerResponse[saperpsharedonto:IDResponse    wsml:hasValue    ?iDResponse]    wsml:memberOf
saperpsharedonto:CustomerResponse and
          ?CustomerResponse[saperpsharedonto:Common      wsml:hasValue      ?common]      wsml:memberOf
saperpsharedonto:CustomerResponse and
          ?common[saperpsharedonto:NameResponse      wsml:hasValue      ?nameResponse]      wsml:memberOf
saperpsharedonto:CommonResponse and
          ?nameResponse[saperpsharedonto:FormOfAddressCode wsml:hasValue ?formOfAddressCode] wsml:memberOf
saperpsharedonto:NameResponse and
          ?nameResponse[saperpsharedonto:FormOfAddressName wsml:hasValue ?formOfAddressName] wsml:memberOf
saperpsharedonto:NameResponse and
          (
           ?nameResponse[saperpsharedonto:FirstLineName   wsml:hasValue   ?firstLineName]   wsml:memberOf
saperpsharedonto:NameResponse and
           ?nameResponse[saperpsharedonto:SecondLineName   wsml:hasValue   ?secondLineName]   wsml:memberOf
saperpsharedonto:NameResponse and
           ?nameResponse[saperpsharedonto:ThirdLineName   wsml:hasValue   ?thirdLineName]   wsml:memberOf
saperpsharedonto:NameResponse and
           ?nameResponse[saperpsharedonto:FourthLineName   wsml:hasValue   ?fourthLineName]   wsml:memberOf
saperpsharedonto:NameResponse
          )
          and
          ?AddressInformation[saperpsharedonto:AddressResponse       wsml:hasValue       ?addressResponse]
wsml:memberOf saperpsharedonto:AddressInformationResponse and
          ?addressResponse[saperpsharedonto:PhysicalAddressResponse                          wsml:hasValue
?physicalAddressResponse] wsml:memberOf saperpsharedonto:AddressResponse and
```

```
          (
           ?physicalAddressResponse[saperpsharedonto:CountryCode wsml:hasValue ?countryCode] wsml:memberOf
saperpsharedonto:PhysicalAddressResponse or
           ?physicalAddressResponse[saperpsharedonto:CountryName wsml:hasValue ?countryName] wsml:memberOf
saperpsharedonto:PhysicalAddressResponse or
           ?physicalAddressResponse[saperpsharedonto:StreetPostalCode   wsml:hasValue   ?streetPostalCode]
wsml:memberOf saperpsharedonto:PhysicalAddressResponse or
           ?physicalAddressResponse[saperpsharedonto:CityName   wsml:hasValue   ?cityName]   wsml:memberOf
saperpsharedonto:PhysicalAddressResponse or
           ?physicalAddressResponse[saperpsharedonto:StreetName  wsml:hasValue  ?streetName]  wsml:memberOf
saperpsharedonto:PhysicalAddressResponse or
           ?physicalAddressResponse[saperpsharedonto:HouseID    wsml:hasValue    ?houseID]    wsml:memberOf
saperpsharedonto:PhysicalAddressResponse
          )
         )
"""^^wsml:AxiomLiteral.
```

Since the WSMO-Lite service model does not prescribe interpretation instruction for the value of `rdfs:Literal` or `wsml:AxiomLiteral`. It is up to an external application or reasoning service to extract these axioms and interpret them according to some application specific setup. The elegance of resource notation in RDF is exemplified by this approach in the WSMO-Lite service model. We have provided for instance in the precondition instance, an axiom as a disjunction of mainly two sub-predicates: the first one is meant to check whether an instance of the domain concept Name has value in its information slots `firstLineName` and `secondLineName` in the service request message instance. It states also that optionally the `PhysicalAddress` concept instance is filled with some meaningful values in the request message instance. If either of these sub-predicates in the axiom is evaluated to true, the axiom is rendered true and thus the precondition instance holds. To reiterate, it is the task of an external reasoning agent to interpret the axiom and evaluate the predicates accordingly.

It is however worth mentioning that there is a caveat that this approach reasonably ignore possible external influence beyond the scope of the respective ES and it information scope. In an enterprise environment, in fact in many other occasions it is perhaps also of interest to logically express the change in the state of the world not just in the information scope of an ES, but also from an overall process point of view. Since the rationale of adopting RDF/S and WSMO-Lite model is pragmatism and there is a tradeoff between high expressivity and high scalability [Lambert2010], we have chosen to rule out the axiomatization from the world state perspective of the domain but refocus just on the enterprise information model and the service functional model for the sake of simplicity, tractability and efficiency.

With this rationale and attempting to consolidate existing WSMO-Lite and MicroWSMO service annotations for SOA4All, the Minimal Service Model [Lambert2010] unifies existing lightweight service annotations in order to provide a basis for building richer set of systematic annotation approach.

3.3 Functional Classification

The WSMO-Lite service model provides the notion of a functional classification root to serve as the starting point of a functional classification structure to categorize services according to a set of user-defined semantic functional descriptions. The root element is defined as shown in the following listing.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
wsl:FunctionalClassificationRoot rdfs:subClassOf rdfs:Class .
```

The notion of the concept `wsl:FunctionalClassificationRoot` is an abstract starting point for a hierarchy of classification concept instances that are attached to this root element. If we think of the classification structure as a search tree where each node represents a functional classification instance, stating some functionality description for a described service instance, we are able to search for a specific functionality according to this classification structure. Each connection arc between two nodes is a link between an upper-level more general classification concept on one end and another more specific concept on the other end. With a series of connection arcs, one can derive a path from the functional classification root down to a specific functional classification concept which refines the functional description of a service. During service search and matchmaking process, this classification structure can be traversed by a service matchmaking agent regarding a certain functional query criteria that is provided, for instance, in a SPARQL query to find a service based on matching on the existing specified functionality classification information to the desired functionality requirement of the query. This is an important and novel principle in lightweight service discovery and matchmaking within the WSMO-Lite model. It implies that not only we can search for a service based on semantic information on input/output message instance, condition and effect in RDF/S but also support search based on functional classification with the advantage of efficient pruning search space within the search tree once we have provided adequate and detailed functional classification information and attached them to our WSMO-Lite service ontology instance.

We show an initial excerpt of our attempt to provide a functional classification of ES structure based on a notion that we call *dimension* in the following listing.

```
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix wsml: <http://www.wsmo.org/wsml/wsml-syntax#>.
@prefix sapesfcmdl:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicefunctionalclassificationmodel#>.
@prefix sapesfcmdli:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicefunctionalclassificationmodelinstance#>.
@prefix sapesfc:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicefunctionalclassification#>.
@prefix sapesfci:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicefunctionalclassificationinstance#>.
@prefix sapesfcr:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicefunctionalclassificationroot#>.
@prefix sapesfcri:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicefunctionalclassificationrootinstance#>.
@prefix sapes: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservice#>.
@prefix sapesi: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseserviceinstance#>.
```



```
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#>.
##############################################################
# process and enterprise service functional classification
##############################################################
sapesfcmdl:BusinessDomainFunctionalClassificationModel a rdfs:Class;
    rdfs:subClassOf sapdmdl:BusinessDomainConceptModel;
    rdfs:subClassOf sapes:SAPEnterpriseServiceFunctionalModel.
sapesfcr:BusinessDomainFunctionalClassificationRoot a rdfs:Class;
    rdfs:comment "Root element of a complete functional classification of processes or enterprise services”;
    rdfs:isDefinedBy sapesfcmdl:BusinessDomainFunctionalClassificationModel;
    rdfs:subClassOf sapbdcp:BusinessDomainConcept;
    rdfs:subClassOf wsl:FunctionalClassificationRoot.
sapesfc:BusinessDomainFunctionalClassificationCategory a rdfs:Class;
    rdfs:subClassOf sapesfcr:BusinessDomainFunctionalClassificationRoot.
sapesfc:BusinessDomainGenericExtensibleFunctionalClassification a rdfs:Class;
    rdfs:comment "Other functional classification dimensions can be defined to extend this concept";
    rdfs:subClassOf sapesfc:BusinessDomainFunctionalClassificationCategory.
sapesfc:BusinessDomainFunctionalDesignationDimension a rdfs:Class;
    rdfs:comment "Designation dimension of functional classification categorizes enterprise service according to ERP software module designation";
    rdfs:subClassOf sapesfc:BusinessDomainFunctionalClassificationCategory.
sapesfc:BusinessDomainFunctionalOperationalDimension a rdfs:Class;
    rdfs:comment "Operational dimension of functional classification categorizes enterprise service according to the operations performed";
    rdfs:subClassOf sapesfc:BusinessDomainFunctionalClassificationCategory.
sapesfc:BusinessDomainFunctionalClassification a rdfs:Class;
    rdfs:isDefinedBy sapesfcmdl:BusinessDomainFunctionalClassificationModel;
    rdfs:comment "Functional classification instance of enterprise service or processes requires definition URI, classification along operational and designational dimension, extension of other dimensions is possible";
    rdfs:subclassOf wsl:FunctionalClassificationRoot;
    rdfs:subClassOf sapesfcr:BusinessDomainFunctionalClassificationRoot;
    sapesfc:hasBusinessDomainFunctionalClassificationNSURI rdfs:Literal;
    sapesfc:hasBusinessDomainOperationalClassification sapesfc:BusinessDomainFunctionalOperationalDimension;
    sapesfc:hasBusinessDomainDesignationClassification sapesfc:BusinessDomainFunctionalDesignationDimension;
    sapesfc:hasBusinessDomainExtensibleFunctionalClassification sapesfc:BusinessDomainGenericExtensibleFunctionalClassification.
###########################################
# classification dimension 1: operational
###########################################
sapesfci:BusinessDomainFunctionalClassificationCreate a sapesfc:BusinessDomainFunctionalOperationalDimension;
    rdfs:comment "This category of enterprise service performs creation operation".
sapesfci:BusinessDomainFunctionalClassificationRead a sapesfc:BusinessDomainFunctionalOperationalDimension;
    rdfs:comment "This category of enterprise service performs read operation".
sapesfci:BusinessDomainFunctionalClassificationSearch a sapesfc:BusinessDomainFunctionalOperationalDimension;
    rdfs:comment "This category of enterprise service performs search operation".
sapesfci:BusinessDomainFunctionalClassificationUpdate a sapesfc:BusinessDomainFunctionalOperationalDimension;
```

```
    rdfs:comment "This category of enterprise service performs update operation".
sapesfci:BusinessDomainFunctionalClassificationDelete a sapesfc:BusinessDomainFunctionalOperationalDimension;
    rdfs:comment "This category of enterprise service performs delete operation".
sapesfci:BusinessDomainFunctionalClassificationIndexing a sapesfc:BusinessDomainFunctionalOperationalDimension;
    rdfs:comment "This category of enterprise service performs index operation".
sapesfci:BusinessDomainFunctionalClassificationReplication a sapesfc:BusinessDomainFunctionalOperationalDimension;
    rdfs:comment "This category of enterprise service performs replication operation".
sapesfci:BusinessDomainFunctionalClassificationHumanActivity a sapesfc:BusinessDomainFunctionalOperationalDimension;
    rdfs:comment "This category of enterprise service performs human task related operation".
##############################################################################
# classification dimension 2: functional designation of process and services
##############################################################################
sapesfci:BusinessDomainFunctionalClassificationERP a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to ERP related functionality".
sapesfci:BusinessDomainFunctionalClassificationPublicSector a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to public sector information
system related functionality".
sapesfci:BusinessDomainFunctionalClassificationGovernmentalService a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to governmental information
system related functionality".
sapesfci:BusinessDomainFunctionalClassificationCRM a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to customer relationship
management related functionality".
sapesfci:BusinessDomainFunctionalClassificationSRM a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to supplier relationship
management related functionality".
sapesfci:BusinessDomainFunctionalClassificationSCM a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to supply chain management
related functionality”.
sapesfci:BusinessDomainFunctionalClassificationManufacturing a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to manufacturing industry
related functionality".
sapesfci:BusinessDomainFunctionalClassificationLogistics a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to logistics business related
functionality".
sapesfci:BusinessDomainFunctionalClassificationFinancial a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to financial and ledging
related functionality".
sapesfci:BusinessDomainFunctionalClassificationAccounting a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to accounting and auditing
related functionality".
sapesfci:BusinessDomainFunctionalClassificationHR a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to human resource management
related functionality".
sapesfci:BusinessDomainFunctionalClassificationBusinessIntelligence
    a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to business intelligence
related functionality".
sapesfci:BusinessDomainFunctionalClassificationBanking a sapesfc:BusinessDomainFunctionalDesignationDimension;
```

```
    rdfs:comment "This category of enterprise service classifies dedication of the service to banking related functionality".
sapesfci:BusinessDomainFunctionalClassificationLegal a sapesfc:BusinessDomainFunctionalDesignationDimension;
    rdfs:comment "This category of enterprise service classifies dedication of the service to legal and legislation related functionality".
```

A *dimension* is used to group an instance of a certain functional classification criteria that we have defined. It serves as a functional tag that can be used to describe a functional classification in one functional aspect, for instance, enterprise processes and applications are often oriented toward CRUD[10]-related functional operations. The above listing is part of the upper-level ontology

sap_esi_layer_business_domain_fundamental_ontology.n3

found in the project repository.

A meaningful functional classification dimension of an ES could therefore be based on such CRUD *operations* that are carried out. In the previous listing, for instance, take the example concept instance

`sapesfci:BusinessDomainFunctionalClassificationCreate`.

It attributes the functional capability of performing a creation operation to the ES and is an instance of a *dimension*. This type of *operational classification* is grouped into an upper category called

`sapesfc:BusinessDomainFunctionalOperationalDimension`,

so that the create dimension of classification is defined as the asserted instance of this RDF/S class.

Another meaningful dimension to classify ES is according to its *designation* in a process model, i.e. for what application domain or module is the service conceived for.

`sapesfc:BusinessDomainDesignationClassificationDimension`

is the RDF/S concept to denote this dimension. Since a discovery and matchmaking agent could query for services that are designated for a specific set of application domains, we have introduced this category as another basic dimension. Our example ES can be classified to the instance of `sapesfci:BusinessDomainFunctionalClassificationCRM` according to the designation dimension.

The key concept of a *functional classification* is the concept

`sapesfc:BusinessDomainFunctionalClassification`

which is a sub-concept of the `wsl:FunctionalClassificationRoot` and hence is attached to and aligned with the WSMO-Lite functional classification model. An instance of this key classification concept possesses an identification URI and a set of operational and designation classification dimension instances that are linked to the `sapesfc:BusinessDomainFunctionalClassification` instance via property slots. We have also devised an additional slot to link an extensible dimension of functional classification called

`sapesfc:BusinessDomainGenericExtensibleFunctionalClassification`

to the instance. It serves as an extension point in principle for further definition of

[10] CRUD is an acronym for the create, read/search, update and delete operations.

classification dimension. Upon a functional classification instance of the concept `sapesfc:BusinessDomainFunctionalClassification`, one can attach a set of instances of dimensions to give functional attribute to a `wsl:Service`. The principle of this quasi multi-dimensional functional classification is shown in Figure 7.

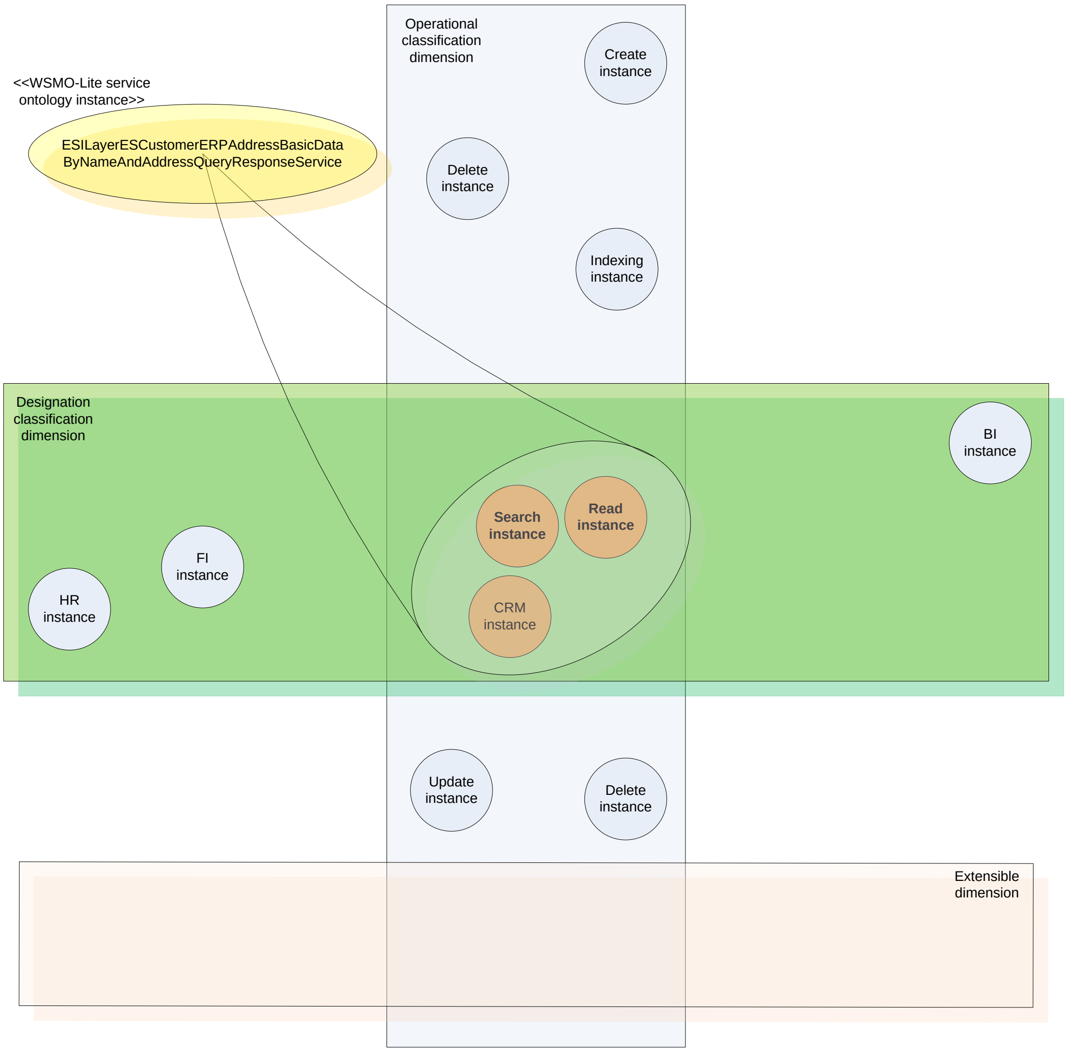

Figure 7: Functional Classification Concept of ES harmonized with WSMO-Lite

Our example ES is depicted with the ellipse showing an instance of WSMO-Lite service ontology described in previous sections. In the following listing we show exactly this conceptual project of the two dimensions: operational and designation with their corresponding functional classification category instances to attribute functional description to our service. This excerpt is found in the WSMO-Lite service ontology instance in file

sap_esi_layer_es_address_basic_data_by_name_and_address_wsmo_lite.n3.

```
 @prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
 @prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#>.
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceFunctionalClassificationInstance
    a sapesfc:BusinessDomainFunctionalClassification;
    sapesfc:hasBusinessDomainFunctionalClassificationNSURI
```

```
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#functionalClassification>;
    sapesfc:hasBusinessDomainOperationalClassification sapesfci:BusinessDomainFunctionalClassificationSearch;
    sapesfc:hasBusinessDomainOperationalClassification sapesfci:BusinessDomainFunctionalClassificationRead;
    sapesfc:hasBusinessDomainDesignationClassification sapesfci:BusinessDomainFunctionalClassificationCRM.
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService a wsl:Service;
    wsl:hasOperation tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation;
    wsl:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#sharedDomainSpecificOntology>;
    wsl:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#commonFundamentalOntology>;
    sawsdl:modelReference
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceFunctionalClassificationInstance;
 rdfs:isDefinedBy
    <http://localhost /ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService?wsdl>.
```

Subsequently in the minimal service model based service ontology instance

msm_sap_esi_layer_es_address_basic_data_by_name_and_address.n3

we have created reference to this classification instance as shown in the following listing.

```
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix msm: <http://cms-wg.sti2.org/ns/minimal-service-model#>.
@prefix refwsmoliteonto: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#>.
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com: erpsv_msm#>.
tns:MSM_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService a msm:Service;
    rdfs:isDefinedBy refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService;
    msm:hasOperation tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation;
    msm:hasInputMessage tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync;
    msm:hasOutputMessage tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync;
    msm:hasOutputFault tns:exception00;
   msm:hasCondition
refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServicePrecondition;
    msm:hasEffect refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceEffect;
    msm:hasFunctionalClassification
refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceFunctionalClassificationInstance.
```

3.4 Minimal Service Model

The minimal service model [Lambert2010] provides a minimal and common conceptual model in RDF/S for capturing the semantics of WSDL and RESTful services by utilizing and consolidating existing WSMO-Lite [Vitvar2008] and MicroWSMO [Vitvar2009] service model for the annotation of both types of services. It provides the ground for treating them homogeneously in service discovery and matchmaking process. It is not an independent new service model at all but a logical extension to the existing mechanisms; while avoiding commitment to the details of the technicalities of the underlying service or service infrastructure as much as possible. The minimal service model also brings in a new mechanism of declarative service request or response notation, called *service templates*, to replace the necessity to denote condition and constraints using expressive logical axioms with all the extra load of expertise required. Instead service templates serve to contribute to more lightweight declarative approach to annotate extra semantic information to a service request/response and allow dynamic generation of SPARQL queries which express conditional constraints encoded within the request or response declaration. It is a novel device to replace the more heavyweight stateful model of conditions as used in WSMO. Service templates will be further discussed with its initial application to our example ES in Chapter 5.

The minimal service model in RDF/S is shown in the following listing.

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix hr: <http://www.wsmo.org/ns/hrests#>.
@prefix msm: <http://cms-wg.sti2.org/ns/minimal-service-model#>.
@prefix st: <http://cms-wg.sti2.org/ns/service-template#>.
##########################
# mimimal service model
##########################
msm:Service a rdfs:Class.
msm:hasOperation a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range msm:Operation.
msm:Operation a rdfs:Class.
msm:hasInputMessage a rdf:Property;
    rdfs:domain msm:Operation;
    rdfs:range msm:Message.
msm:hasOutputMessage a rdf:Property;
    rdfs:domain msm:Operation;
    rdfs:range msm:Message.
msm:hasInputFault a rdf:Property;
    rdfs:domain msm:Operation;
    rdfs:range msm:Message.
msm:hasOutputFault a rdf:Property;
    rdfs:domain msm:Operation;
    rdfs:range msm:Message.
msm:Message a rdfs:Class.
msm:usesOntology a rdfs:Property;
    rdfs:domain msm:Service;
    rdfs:subPropertyOf rdfs:seeAlso.
msm:hasFunctionalClassification a rdfs:Property;
    rdfs:subPropertyOf sawsdl:modelReference.
msm:hasNonfunctionalProperty a rdfs:Property;
```

```
    rdfs:subPropertyOf sawsdl:modelReference.
msm:hasCondition a rdfs:Property;
    rdfs:subPropertyOf sawsdl:modelReference.
msm:hasEffect a rdfs:Property;
    rdfs:subPropertyOf sawsdl:modelReference.
#############################
# service template
#############################
st:ServiceTemplate a rdfs:Class.
st:hasFunctionalClassification a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:hasInput a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:hasOutput a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:hasPreference a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:hasRequirement a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:value a rdf:Property;
    rdfs:domain rdfs:Resource;
    rdfs:range rdfs:Datatype.
#####################################
# service template match reporting
#####################################
st:MatchingResults a rdfs:Class.
st:MatchDegree a rdfs:Class.
st:ExactMatch a st:MatchDegree.
st:PluginMatch a st:MatchDegree.
st:SubsumesMatch a st:MatchDegree.
st:matches a rdf:Property;
    rdfs:domain st:ServiceTemplate;
    rdfs:range msm:Operation.
st:matchDegree a rdf:Property;
    rdfs:domain st:ServiceTemplate;
    rdfs:range st:MatchDegree.
###########################
# WSMO-Lite consolidation
###########################
```

```
wsl:Ontology rdfs:subClassOf owl:Ontology.
wsl:FunctionalClassificationRoot rdfs:subClassOf rdfs:Class.
wsl:NonFunctionalParameter a rdfs:Class.
wsl:Condition a rdfs:Class.
wsl:Effect a rdfs:Class.
###############################################
# hRESTS consolidation
###############################################
hr:hasAddress a rdf:Property;
   rdfs:domain msm:Operation;
 rdfs:range hr:URITemplate.
hr:hasMethod a rdf:Property;
 rdfs:domain msm:Operation;
 rdfs:range xsd:string.
hr:URITemplate a rdfs:Datatype.
######################
# SAWSDL properties
######################
sawsdl:modelReference a rdf:Property.
sawsdl:liftingSchemaMapping a rdf:Property.
sawsdl:loweringSchemaMapping a rdf:Property.
```

Since our example ES is WSDL based and we have shown essential excerpts of the WSMO-Lite service ontology, we proceed with creating our minimal service ontology called

msm_sap_esi_layer_es_address_basic_data_by_name_and_address.n3

which can be found, together with corresponding WSMO-Lite ontology in the project svn repository:

http://svn.sti2.at/soa4all/trunk/soa4all-usecases/WP7/ontology/D7.6

We show the essential elements of our minimal service model service ontology in the following listing.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix wsml: <http://www.wsmo.org/wsml/wsml-syntax#>.
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix msm: <http://cms-wg.sti2.org/ns/minimal-service-model#>.
@prefix st: <http://cms-wg.sti2.org/ns/service-template#>.
@prefix refwsmoliteonto: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#>.
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com: erpsv_msm#>.
tns:MSM_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceOntology
  a wsl:Ontology;
    rdfs:comment "MSM ontology (CustomerERPAddressBasicDataByNameAndAddressQueryResponse_InService)";
    rdfs:isDefinedBy
```

```
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#sharedDomainSpecificOntology>;
    rdfs:seeAlso
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#sharedDomainSpecificOntology>;
    rdfs:isDefinedBy
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#commonFundamentalOntology>;
    rdfs:seeAlso <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#commonFundamentalOntology>;
    rdfs:isDefinedBy <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#
WSMOlite_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceOntology>;
    rdfs:seeAlso <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#
WSMOlite_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceOntology>.
#############################################################################################
# declaration of messages, operation and functional classification of MSM service instance
#############################################################################################
tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync a msm:Message;
    rdfs:subClassOf refwsmoliteonto:CustomerERPAddressBasicDataByNameAndAddressQuery_sync.
tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync a msm:Message;
    rdfs:subClassOf refwsmoliteonto:CustomerERPAddressBasicDataByNameAndAddressResponse_sync.
tns:exception00 a msm:Message;
    rdfs:subClassOf refwsmoliteonto:exception00.
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation a msm:Operation;
    rdfs:subClassOf refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation.
tns:MSM_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService a msm:Service;
   rdfs:comment "This instance references the declared WSMO-Lite service instance";
   rdfs:isDefinedBy refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService;
   msm:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#commonFundamentalOntology>;
   msm:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#sharedDomainSpecificOntology>;
   msm:usesOntology <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite
        #WSMOlite_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceOntology>;
   msm:hasOperation tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation;
   msm:hasInputMessage tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync;
   msm:hasOutputMessage tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync;
   msm:hasOutputFault tns:exception00;
   msm:hasCondition refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServicePrecondition;
   msm:hasEffect refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceEffect;
   msm:hasFunctionalClassification
refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceFunctionalClassificationInstance.
```

The minimal service model service ontology instance begins with a declaration of itself as highlighted in the first line after the namespace prefixes. Instead of re-inventing the wheel here and to reuse the WSMO-Lite service ontology, we have declared a namespace prefix `refwsmoliteonto` to qualify the namespace of the elements of the WSMO-Lite ontology instance when necessary. Both the input/output message and the outbound fault message instance are semantically linked back to the corresponding concept instances within the WSMO-Lite ES ontology. This is also true for the operation instance

ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation that is used subsequently in the service property slots. The service element

MSM_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService

is declared as an instance of the `msm:Service` concept with its information slots showing extensive linkages with the underlying elements in our existing WSMO-Lite service ontology. It is worth noticing that the property `msm:usesOntology` provides the service model a way to specify necessary and relevant set of ontology that is used by the current one and states the dependency of this ontology with others.

For compatibility with the WSMO-Lite service model, it is possible to link in our defined precondition instance and effect instance using the properties `msm:hasCondition` and `msm:hasEffect`.

For the above example list, reader notices in the last line the usage of the property `msm:hasFunctionalClassification` to provide a linkage of our declared functional classification instance to the `msm:Service` instance.

4. Service Annotation with Non-Functional Properties

Similar to the functional classification described in the previous section, *contextual information* about a service also includes non-functional description of a service. We choose to declare functional dimensions to cluster functional properties with the previously described classification approach. For non-functional properties are all those attributes to refine a service description that are not functional by nature, for instance, attempting to get attributive description of the response time or down-time of a service as a performance indicator in order to use these indicator for service selection are examples of non-functional properties of ES.

4.1 Basic Non-Functional Properties

Since the scope of non-functional properties in enterprise application domain is broad and an exhaustive enumeration of all available non-functional parameters that are relevant for ES is prohibitive, we have exemplarily defined in the fundament upper-level ontology a limited set of non-functional parameters to refine description of our service. In the following listing we show the conceptualization of functional parameter in our domain.

```
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix wsml: <http://www.wsmo.org/wsml/wsml-syntax#>.
@prefix sapesnfpmdl:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicenonfunctionalparametermodel#>.
@prefix sapesnfp:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicenonfunctionalparameter#>.
@prefix sapesnfpi:
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:enterpriseservicenonfunctionalparameterinstance#>
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#>.
##########################################################################
# process and enterprise service non-functional paramter (NFP) aspects
##########################################################################
sapesnfpmdl:BusinessDomainNonFunctionalCharacterisationModel a rdfs:Class;
    rdfs:subClassOf sapdmdl:BusinessDomainConceptModel;
    rdfs:subClassOf sapes:SAPEnterpriseServiceFunctionalModel.
sapesnfp:BusinessDomainNonFunctionalParameter a rdfs:Class;
    rdfs:isDefinedBy sapesnfpmdl:BusinessDomainNonFunctionalCharacterisationModel;
    rdfs:comment "Every instance of non-function parameter prescribes a parameter value and a unit for
interpretation";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI rdfs:Literal;
    sapesnfp:hasNFPvalue sapesnfp:BusinessDomainNonFunctionalParameterMagnitude;
    sapesnfp:hasNFPvalueUnit sapesnfp:BusinessDomainNonFunctionalParameterUnit;
    rdfs:subClassOf wsl:NonFunctionalParameter.
sapesnfp:BusinessDomainNonFunctionalParameterMagnitude a rdfs:Datatype;
    rdfs:isDefinedBy sapesnfpmdl:BusinessDomainNonFunctionalCharacterisationModel.
```

```
sapesnfp:BusinessDomainNonFunctionalParameterUnit a rdfs:Literal;
    rdfs:isDefinedBy sapesnfpmdl:BusinessDomainNonFunctionalCharacterisationModel.
sapesnfp:hasNFPvalue a rdf:Property;
    rdfs:domain sapesnfp:BusinessDomainNonFunctionalParameter;
    rdfs:range sapesnfp:BusinessDomainNonFunctionalParameterMagnitude.
sapesnfp:hasNFPvalueUnit a rdf:Property;
    rdfs:domain sapesnfp:BusinessDomainNonFunctionalParameter;
    rdfs:range sapesnfp:BusinessDomainNonFunctionalParameterUnit.
sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI a rdf:Property;
    rdfs:domain sapesnfp:BusinessDomainNonFunctionalParameter;
    rdfs:range rdfs:Literal.
##################################################
# NFP – a sub-category of performance related NFPs
##################################################
sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP a rdfs:Class;
    rdfs:comment "Definition of concepts of performance related non-functional parameters";
    rdfs:isDefinedBy sapesnfp:BusinessDomainNonFunctionalParameter;
    rdfs:subClassOf sapesnfp:BusinessDomainNonFunctionalParameter.
###############################################################################
# NFP – a sub-category of SLA-related NFPs (other NFP can be added if necessary)
###############################################################################
sapesnfp:SAPEnterpriseServiceSLArelatedNFP a rdfs:Class;
    rdfs:comment "Defintion of concepts of SLA (service level agreement) related non-functional parameter";
    rdfs:isDefinedBy sapesnfp:BusinessDomainNonFunctionalParameter;
    rdfs:subClassOf sapesnfp:BusinessDomainNonFunctionalParameter.
```

The core concept of non-functional parameter for ES is defined with `sapesnfp:BusinessDomainNonFunctionalParameter`.

It is a sub-concept of `wsl:NonFunctionalParameter` of the WSMO-Lite service model. An instance of this concept has an identification URI for namespace qualification. Two information slots are defined to give this concept contextual interpretation based on RDF-property-linkage to instance of these concepts:

- `sapesnfp:BusinessDomainNonFunctionalParameterMagnitude` and
- `sapesnfp:BusinessDomainNonFunctionalParameterUnit`.

The first concept confers a magnitude to the non-functional parameter, thus allowing a contextual interpretation of *mass* of the non-functional parameter. The second concept is important because we can often interpret a certain given magnitude only with an appropriate *unit* which serves to allow a *contextual interpretation* of the *mass* based on mapping the *mass* to an *interpretation domain*.

We have defined the magnitude concept to take `rdfs:Datatype` and the unit concept to take `rdfs:Literal` to allow flexibility of supporting and expressing arbitrarily defined sub-concept of `sapesnfp:BusinessDomainNonFunctionalParameter` later. For our example ES we have defined two sub-concepts mainly for grouping non-functional parameters.

- `sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP` and

- sapesnfp:SAPEnterpriseServiceSLArelatedNFP

The first concept category allows non-functional parameter instances to be clustered to performance related characteristics such as service latency or down-time. The second concept category is related to service-level-agreement aspects such as pricing model attributes or licensing issues. As a matter of fact our initial attempt in this regard is shown in the following listing with possibility of further extension if required.

```
######################################################################
# performance related NFP instances (in relation with WSMO-Lite NFP)
######################################################################
sapesnfpi:NFPprocessingLatencyMinimal a sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP;
    rdfs:comment "Processing latency of an enterprise service is defined as minimal if it does not exceed 0.2 millisecond";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:processingLatency;
    sapesnfp:hasNFPvalue "0.2"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "millisecond"^^xsd:string.
sapesnfpi:NFPprocessingLatencyMinimal a wsl:NonFunctionalParameter.
sapesnfpi:NFPprocessingLatencyMedium a sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP;
    rdfs:comment "Processing latency of an enterprise service is defined as medium if it does not exceed 10 milliseconds";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:processingLatency;
    sapesnfp:hasNFPvalue "10.0"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "millisecond"^^xsd:string.
sapesnfpi:NFPprocessingLatencyMedium a wsl:NonFunctionalParameter.
sapesnfpi:NFPprocessingLatencyHigh a sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP;
    rdfs:comment "Processing latency of an enterprise service is defined as high if it exceeds 50 milliseconds";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:processingLatency;
    sapesnfp:hasNFPvalue "50.0"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "millisecond"^^xsd:string.
sapesnfpi:NFPprocessingLatencyHigh a wsl:NonFunctionalParameter.
sapesnfpi:NFPprocessingDowntimeMinimal a sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP;
    rdfs:comment "Availability of enterprise service is defined as the amount of downtime";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:processingDowntime;
    sapesnfp:hasNFPvalue "0.001"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "hours"^^xsd:string.
sapesnfpi:NFPprocessingDowntimeMinimal a wsl:NonFunctionalParameter.
sapesnfpi:NFPprocessingDowntimeAcceptable a sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP;
    rdfs:comment "Acceptable amount of downtime is defined as 0.05 hours per year";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:processingDowntime;
    sapesnfp:hasNFPvalue "0.05"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "hours"^^xsd:string.
sapesnfpi:NFPprocessingDowntimeAcceptable a wsl:NonFunctionalParameter.
sapesnfpi:NFPprocessingDowntimeHigh a sapesnfp:SAPEnterpriseServicePerformanceRelatedNFP;
    rdfs:comment "High downtime and thus low availability is any service with downtime more than 0.5 hours per year";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:processingDowntime;
```

```
    sapesnfp:hasNFPvalue "0.5"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "hours"^^xsd:string.
sapesnfpi:NFPprocessingDowntimeHigh a wsl:NonFunctionalParameter.
############################################################
# SLA related instances (in relation with wsmo-lite NFP)
############################################################
sapesnfpi:NFserviceUsageCostFree a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "Usage of the enterprise service is free of charge";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "0"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "Euro"^^xsd:string.
sapesnfpi:NFserviceUsageCostFree a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceUsageCostLow a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "Usage cost of the enterprise service is believed to be low";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "0.5"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "Euro"^^xsd:string.
sapesnfpi:NFserviceUsageCostLow a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceUsageCostMedium a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "Usage cost of the enterprise service is believed to be medium";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "2"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "Euro"^^xsd:string.
sapesnfpi:NFserviceUsageCostMedium a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceUsageCostHigh a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "Usage cost of the enterprise service is believed to be high";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "10"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "Euro"^^xsd:string.
sapesnfpi:NFserviceUsageCostHigh a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceUsageCostPremium a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "Usage cost of the service is believed to be high and belongs to the premium level";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "50"^^xsd:long;
    sapesnfp:hasNFPvalueUnit "Euro"^^xsd:string.
sapesnfpi:NFserviceUsageCostPremium a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceLicenseModeFreeUsage a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "License mode allows free usage of an enterprise service";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "usage license is free with unlimited usage";
    sapesnfp:hasNFPvalueUnit "textual description"^^xsd:string.
sapesnfpi:NFserviceLicenseModeFreeUsage a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceLicenseModePayPerUsage a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "License mode allows pay-per-usage based of enterprise service consumption";
```

```
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "usage license is based on pay per individual usage"^^xsd:string;
    sapesnfp:hasNFPvalueUnit "textual description"^^xsd:string.
sapesnfpi:NFserviceLicenseModePayPerUsage a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceLicenseModeSubscriptionBasedUsage a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "License mode allows service consumption to be linked to a subscription constract";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "usage license is coupled with subscription contract"^^xsd:string;
    sapesnfp:hasNFPvalueUnit "textual description"^^xsd:string.
sapesnfpi:NFserviceLicenseModeSubscriptionBasedUsage a wsl:NonFunctionalParameter.
sapesnfpi:NFserviceLicenseModeBundlePurchaseBasedUsage a sapesnfp:SAPEnterpriseServiceSLArelatedNFP;
    rdfs:comment "License model allows unlimited consumption by specific purchased software bundle";
    sapesnfp:hasBusinessDomainNonFunctionalParameterNSURI sapesnfpi:slaUsageCost;
    sapesnfp:hasNFPvalue "usage license is obtained automatically on software purchase"^^xsd:string;
    sapesnfp:hasNFPvalueUnit "textual description"^^xsd:string.
sapesnfpi:NFserviceLicenseModeBundlePurchaseBasedUsage a wsl:NonFunctionalParameter.
```

The previous listing is an excerpt from the upper-level ontology found in the file:

sap_esi_layer_business_domain_fundamental_ontology.n3

Interpretation of the magnitude and unit concept is model agnostic in our attempt, i.e. we do not enforce concrete semantics about this parameter definitions nor do we enforce the contextual interpretation or the reasoning process needed to reason about this *contextual information*. For the sake of flexibility, application domain or concrete business settings are able to decide on the concrete interpretation of this information.

Finally we show in the following list the application of the defined non-functional parameter concept instances to our WSMO-Lite service ontology.

```
tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService a wsl:Service;
    wsl:hasOperation tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation;
    wsl:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#sharedDomainSpecificOntology>;
    wsl:usesOntology
<urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:upper_onto#commonFundamentalOntology>;
    wsl:usesOntology
tns:WSMOlite_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceOntology;
    sawsdl:modelReference
<http://sap.com/xi/APPL/SE/Global/WSMO-
Lite/ontology#ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService>;
 rdfs:comment "The next 4 lines includes four non-functional parameters to the service";
 sawsdl:modelReference sapesnfpi:NFserviceUsageCostFree;
 sawsdl:modelReference sapesnfpi:NFserviceLicenseModeFreeUsage;
 sawsdl:modelReference sapesnfpi:NFPprocessingLatencyMinimal;
 sawsdl:modelReference sapesnfpi:NFPprocessingDowntimeMinimal;
 rdfs:isDefinedBy
<http://localhost /ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService?wsdl>.
```

Subsequently the non-functional parameters are referenced also in the minimal service model service ontology instance as shown in the following listing.


```
tns:MSM_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService a msm:Service;
    rdfs:isDefinedBy refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService;
    msm:hasOperation tns:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponsePTOperation;
    msm:hasInputMessage tns:CustomerERPAddressBasicDataByNameAndAddressQuery_sync;
    msm:hasOutputMessage tns:CustomerERPAddressBasicDataByNameAndAddressResponse_sync;
    msm:hasOutputFault tns:exception00;
    msm:hasCondition
refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServicePrecondition;
    msm:hasEffect refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceEffect;
    msm:hasNonfunctionalProperty sapesnfpi:NFserviceUsageCostFree;
    msm:hasNonfunctionalProperty sapesnfpi:NFserviceLicenseModeFreeUsage;
    msm:hasNonfunctionalProperty sapesnfpi:NFPprocessingLatencyMinimal;
    msm:hasNonfunctionalProperty sapesnfpi:NFPprocessingDowntimeMinimal;
    msm:hasFunctionalClassification
refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceFunctionalClassificationInstance.
```

While the WSMO-Lite service model directly embeds the non-functional parameter instances using SAWSDL, the minimal service model service ontology instance uses an RDF-property information slot to link in the non-functional parameters.

4.2 Business Semantics

After studying the requirements of the public sector domain for a service delivery platform based on SOA4All (also see previous Deliverables D7.1, D7.2, D7.3 and D7.4), we identified the importance of supporting civil servants during service selection and administrative process modeling: Civil servants (and other business users) are primarily not interested in the technical details of a Web service such as the definition of parameters (which is often the focus in service annotations) but on the business aspects associated with that service, e.g., the pricing model, the legal implications, the service quality, the security level etc. Traditionally, in economics such different parameters are denoted as Key Performance Indicators (KPIs) that can be measured and analyzed.

In the following, we

- introduce a novel approach to describe business semantics in a dedicated service description language, the Unified Service Description Language (USDL), complete with a WSML ontology to be used in SOA4All
- describe a novel approach for selecting services based on the specific, context-dependent business requirements of a SOA4All user during service invocation and service composition (process modeling)
- provide sample annotations of business semantics in USDL/WSML for selected SAP Enterprise Services (ES) and 3rd party services for the WP7 scenario previously described in Deliverables D7.2 and D7.3

4.2.1 Requirements by the Public Sector

In a rapidly changing society and the age of financial crisis, the public sector is facing a number of challenges, forcing public administration to “provide better, faster and sometimes even more services”. These changes include socio-economic and environmental changes like demographic movements, globalization and internationalization as well as political changes, which lead to reluctance to put more money to the public sector and growing

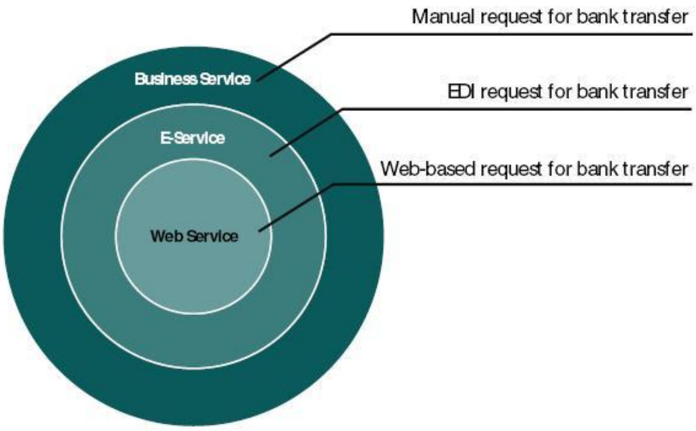

Figure 8: Different Types of Services

mistrust to political and administrative systems. All these factors originated a series of public administration reforms in the last two decades [Ehm2009].

The trend to adopt successful theories from the private sector and to steer public organizations like private ones, is often labeled "New Public Management". The use of advanced Information Technology for the remedy of the above mentioned problems is known as eGovernment [Ehm2009]. Its main purpose is to "transform the processes in which public services are generated and delivered, thereby transforming the entire range of relationships of public bodies, by using the Information Society Technology as enablers" [Leitner2003]. The "new public sector organisation" arose from the "marriage of administrative reform" with "business type managerialism" [Hood1991]. Adopting models to the public sector, which contain output oriented control and quantified performance goals, friendly, effective and efficient service delivery to the constituents, an introduction of performance management took place [Ehm2009]. Embracing concepts like performance management, quality management, decentralization of resources, separation of policy and administration leads to a competitive market of public services [Becker2009].

The existence of public administration organizations is justified primarily by their mission, which is to "serve citizens to advance the common good" [Denhardt2007]. The production of the administration processes is the core task of public administration [Becker2009] and the real value for their customers. Therefore clear business processes are one essential prerequisite for the successful implementation of eGovernment. This means that future administration offices need to leave the 'silo-oriented' organization structure and move to a process oriented organization with the goal to provide qualitative and efficient service to its customers [Ziekow2009]. Administration activities are then considered as output-oriented combined value-adding actions, which should be completed in a certain period of time [Becker2009]. At the same time, process reorganization and reengineering lead to structural organization changes and to redistribution of responsibilities. An effective administration network as a new organizational form has to be established, where clearly defined responsibilities are distributed beyond the organizational borders [Ziekow2009].

A specific implementation of the above mentioned visions in the European Union is the EU Services Directive (also see D7.2). It targets at facilitating and harmonizing the provisioning of services within the EU, where "service" in this context means all sorts of economic services and includes consulting, construction, maintenance, advertising, tourism, etc. The Directive's vision is "to make progress towards a genuine Internal Market in Services so that, in the largest sector of the European economy, both businesses and consumers can take full advantage of the opportunities it presents. To reach these goals, the Services Directive implies a number of general business requirements like (1) establishing electronic procedures and electronic document exchange than can be provided and consumed via the Internet, (2) faster and limited process execution time, (3) better reuse of services and processes etc (please refer to D7.2 and D7.3 for a more detailed discussion).

To fulfill such efficiency and flexibility requirements, an appropriate IT architecture for public administrations is needed. With respect to the IT, the EU Services Directive requires public administrations to integrate data and services from different sources ranging from freely available to enterprise-level services, also across organizational and national boundaries. In addition, the single point of contact needs to handle all sorts of administrative processes that have to be adapted to the specific cases, e.g., when helping a service company to set up offices in two different cities. Moreover, an advanced eGovernment will lead to more empowered constituents who determine independently which services from which governmental and private organizations they need in a particular situation. A great challenge for such distributed systems is the ability to react rapidly and efficiently to a changing environment. The most frequent changes are alignment of business processes to the business goals, the organizational structure or due to possibility to improve the offered services. Besides, changes from outside the organisation like e.g. legislation, social and political upheaval, the large number of public services and the dependencies between them, make an efficient process reconfiguration pretty complex [Lytras2006]. Therefore the ability to provide public services, which best meet the overall requirements, is a key prerequisite for an accurate and effective eGovernment.

In the following sections, we therefore outline a novel approach for supporting civil servants (business users) in modeling and adapting administrative procedures (business processes) by selecting services (process activities) that fit their specific requirements (Key Performance Indicators KPIs). Technically, this is realized by describing KPIs on the one hand and service properties on the other hand semantically so that the service selection during process modeling can be automated to a certain degree. First, we introduce the USDL, a novel semantic notation for describing business aspects of semantic Web services.

4.2.2 Business Semantics for Semantic Web Services

So far, research on Service Oriented Computing has concentrated on software applications, constructed by composing and configuring eServices. SOA and Web services have mainly served as technological solutions that enable enterprise functionality to be made available to users as shared and re-usable services on a network. Traditional metadata that describes services intended for these application integration purposes is based on specification languages that were developed in combination with the early service-based system architectures (SOA and Web services). These languages, e.g., WSDL and BPEL, only target the description of technical characteristics of services [IoS2010]. However, a key aspect has been dramatically discarded in recent research, namely the fact that the main goal of an eService is to provide access to a business service, which may need a radically different description than the eService [DiNitto2009]. For SOA4All`s vision to integrate the service world of large enterprises, SMEs, and end-users, enabling them to engage as peers (i.e. service consumers as well as service providers) within a network of equals it is not sufficient to describe the functional and non-functional technical aspects of eServices. Far more, an enhanced service representation of generic (business) services is needed that captures and aligns business, operational and technical characteristics [IoS2010].

But what are actually services? The terms eService, Service and Web Service are often used as synonyms, though referring to different concepts (see Figure 8). We adopt the service definition introduced by [Cardoso2008]. In business science, the term service is used as the non-material equivalent of a good. Business services are business activities provided by a service provider to a service consumer to create a value for the consumer. In traditional industries they are discovered and invoked manually, but they can be delivered automatically or by humans. A subset of the business services represent the e-services, which are “collections of network-resident software services accessible via standardized protocols, whose functionality can be automatically discovered and integrated into applications or composed to form more complex services”. eServices made available for consumers using Web-based protocols are called Web services [Cardoso2008]. In the SOA4All context, business services are provided to ordinary end users, who might or might not be (technical or business) experts. The vision is to have an open, dynamic and ad-hoc service environment, where billions of parties expose, combine and consume large numbers of services.

With the move to the Internet of Services a new way for service discovery and invocation has emerged (also see D7.2). The Internet is used as a medium for offering and selling services, i.e. treating services as tradable goods [Cardoso2008]. In the Internet of Services, service platforms enable automatic service discovery, provide a unique service description, service composition and negotiation, as well as QoS-based service level agreements and access rights handling. In order to provide an enhanced service description, which meets the requirements of the Internet of Services, a conceptual model for the capture of business-related service data is required. Recently, SAP introduced a novel service description language for that purpose, the so-called Unified Service Description Language (USDL)[11]. The USDL is based on the insights gained in several publicly-funded research projects (incl. SOA4All) as well as on SAPs general knowledge of business services and processes in different industries and application areas. The first version of the USDL was presented at the Future Internet Assembly (FIA) in Stockholm in 2009. Also, a public Web site is available at http://www.internet-of-services.com/.

USDL builds on models for describing business and technical services, and creates a unified description of related research efforts. The purely business description of services has been driven by research on the E3Service ontology, PAS 1018 and the taxonomy of non-functional properties of services identified by O’Sullivan [OSullivan2006]. From the technical side, the most significant proposals to describe services that have influenced USDL include WSDL, WSMO, and OWL-S. Additionally, USDL introduces a new dimension called the operational perspective. This perspective acquires a special importance when several participants are involved in the provision of a service. It should be pointed out that USDL is not meant to replace other specifications in the technical service stacks, but aims to complement them by adding essential business information required for the interaction between service consumer and service provider (possibly involving additional roles such as service hoster, service broker etc.). On the other side, the USDL was not designed to target automated services only but is generic enough to be used for the description of manual services that have no technical implementation. The general design principle was to create a unifying entry point into the overall set of service metadata, which in the end comprises several artifacts in different formats [IoS2010]. Bearing in mind the technical requirement of SOA4All for a directory of semantically annotated processes and services, USDL represents a good alternative for bringing semantic information to the service description.

[11] See USDL section of http://www.internet-of-services.com/

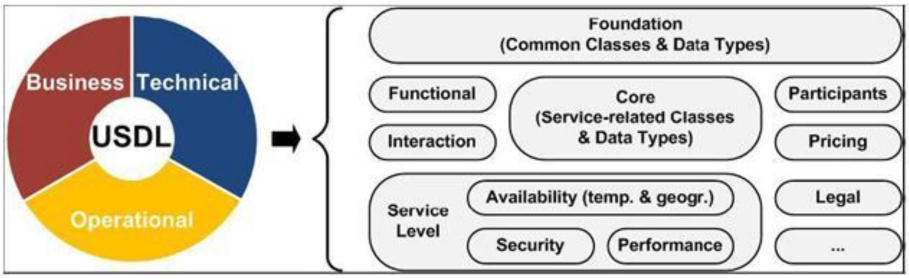

Figure 9: USDL Modules

4.2.3 General Structure of the USDL[12]

The USDL defines a model to describe services in the Internet of Services and services computing in general. Capturing relevant concepts in a structured way is the main challenge of any model design and while designing the USDL, three general groups of concepts were identified:

- Foundational concepts: These concepts cannot be associated uniquely with a single aspect of service description. They capture common entities that occur in the context of different aspects. Examples include time and location concepts, as well as organizational concepts. The foundation also comprises general characteristics common among entities, e.g. name, description, or unique identifiers.
- Generic service description (horizontal) concepts: These concepts apply to a broader range of services without being specific to an application or a domain that exists in the Internet of Services. Examples include concepts like pricing, service availability, or legal parameters.
- Domain-specific service description (vertical) concepts: These concepts capture description aspects that are rather special, i.e., are particular to a certain application or industry vertical.

The USDL is conceived as a generic language and therefore covers concepts from the first two groups. Even though the USDL is thereby limited to the generic part of service description, the set of concepts to model is still quite extensive. In order to ease readability and simplify maintenance of the model, it was decided to separate it into subsets of concepts that are logically related. Relatedness in this context means: “being primarily concerned with the same aspect of service description”. Thus, subsets, henceforth called modules, are aligned with the different aspects identified.

Figure 9 shows the set of modules that are currently conceptualized in USDL. Among them, the *USDL Foundation* is regarded as a special part and usually not called module, but just foundation. It contains elements that belong to the first group of concepts listed earlier, i.e. foundational concepts. All other modules depend on the foundation, which means they refer to elements from it. Another noteworthy part is the *Core* module. It contains the concepts that represent the entities that USDL is actually describing, namely services and service bundles. This module basically ties together all aspects of service description distributed across the other modules.

The formalism used to represent the USDL model is eCore[13], where service description

[12] Please note that the USDL is work in progress and is constantly being improved. In this document, we therefore describe the current status of the USDL, version 3.0

concepts are defined as eCore classes, having typed attributes and associations to other classes. eCore represents the metamodel of the Eclipse Modeling Framework (EMF) and can be considered a dialect of UML.

With the choice of eCore, or UML in general, there are two principal ways of implementing the conceptual modularization: One way is to put all concepts into a single package and define aspect-specific views, which constitute the modules. This would mean that the model is not actually separated and design would have to be highly synchronized. The other way is to create one package per module and reference concepts from other modules through package import. This allows slight decoupling of the design process, as people can work on packages independently and only have to synchronize on the touch points between packages. Given the complexity of USDL, it seemed natural to choose the second option.

Because of the use of the WSMO-Lite ontology for the needs of SOA4All, the USDL eCore classes were conceptualized as a WSML ontology, which can be easily attached to the already annotated services. Adopting the idea from above, the different modules were separated each in a single ontology and imported where needed.

4.2.3.1 USDL-Foundation

The foundation contains generic concepts that are reused in different modules such as the artifact concept, which represents an external entity that can be linked to a USDL element (concept) such as a WSDL document (Functional module), a file defining the terms of use of the service (Legal module), or a piece of documentation for the service (Core module). Resource is another generic concept that is defined in the foundation and represents real-world objects of various types that either performs the service or upon which the service acts, e.g. an application, a system, a tool, an employee, etc.

Another important set of elements provides means to identify, name and generically describe other USDL elements. Unifying the latter two, the element description concept is probably the most widely re-used element of USDL. Its objective is to provide an advanced and fairly complete solution to describe the entities of the USDL as precisely as possible.

In addition, the foundation defines some concepts to define locations as this is needed in various modules for example to define the delivery and availability location of a certain service, or to define a specific price for certain locations, or to specify certain attributes of the users and organizations that have to do with the service.

Furthermore, the foundation defines some concepts to express time as this is also needed in various modules to express for instance quality of service and service level aspects or the availability time of the service or to define the validity period of prices.

4.2.3.2 Core Module

The Core module can be regarded as the center of USDL. It ties together all the aspects of service description that are distributed across other USDL modules. At the heart of the Core module are concepts that represent the entities that USDL is actually describing, namely entities provisioned into service networks (e.g. services, service bundles). It is important to understand that in this context only network level aspects of service description are of interest. Agents active in a service network need to communicate and understand aspects including:

- Service Release: What is the service? Who is able to use a service for which purposes (consumption, aggregation, channeling, etc.)? What are the conditions for

[13] http://www.eclipse.org/modeling/emf/?project=emf

usage?

- Service Access: How, where and when is a service made available?
- Service Consumption: How is a service successfully consumed? What is where and when expected by different parties participating in service provisioning/delivery before, during and after the execution of the service?

Out of scope of USDL are aspects only relevant before the service is provisioned into the network, e.g. within the organization that provides the service. For example, it is neither considered how a service is developed, nor how its quality is assured internally.

Network provisioned entities are detailed along various dimensions of typing, i.e. general service nature, functional granularity and decomposition. Out of these three types, decomposition had the most profound impact onto the model. By default, a service is treated as an indivisible (atomic) entity, meaning it is not subject to decomposition, even though its internal implementation may be arbitrary complex. A composite service, in turn, is a specialization of this concept, as it consists of other components that are known in the ecosystem of the application processing the service description. In order to allow for flexible composition, a component may take the form of a concrete service (atomic or composite), a service bundle or an abstract service. All three of them implement an interface (Composable) for this purpose. Composite services may impose order constraints upon their components in the form of a simple process definition. Please note that the USDL does not replace a fully-fledged process language such as the LPML defined in WP6.

The concept of abstract services is used to define templates that describe classes of services. The idea is to pre-set certain service aspects so that an abstract service may serve as a placeholder for concrete services. The placeholder can then be bound to a concrete service at some point in time (e.g. later than design time).

Another important design decision was to treat functionality-based and business-based decomposition as two separate concepts. This led to the division of tradable entities into service and service bundle. The main reason is that a composite service combines individual services to provide new capabilities, i.e. the component services work together to achieve a higher function. A service bundle on the other hand only combines services for commercial reasons, e.g. competitive pricing or increase of sales numbers. The services are functionally independent and do not influence each other in any way. In contrast, services aggregated in a composition usually have dependencies between each other and may even be aware that they are executed in the context of a composition.

The service nature generally distinguishes automated, semi-automated and human services. Adding the aspect of granularity, i.e. functional complexity, to this, services can be characterized as either providing an elemental function or a more comprehensive set of capabilities. The latter may involve a resource that is managed or exposed by the service.

Common among all top-level service concepts (service, service bundle, abstract service) is that they may have dependencies with other entities. This does not always mean that one service depends on another service. In fact, quite a number of different dependencies can be observed. Services may require another service or a resource. One service may enhance another one, i.e. provide additional capabilities, or may be suitable to substitute another service although its functionality is quite different. The dependency concept captures the dependencies that services, service bundles or abstract services have.

4.2.3.3 Participants Module

The provisioning, trade, delivery and consumption of services and service bundles through service networks potentially involve a multitude of actors. The one actor that holds governance and operational responsibility for a service is commonly referred to as service provider. It controls how the service is provisioned to consumers, e.g. what are the organizational and system resources used, or how it is implemented. In most cases the provider will also act as the trading partner to consumers and define business aspects of delivery. However, there are scenarios in which this function is performed by another legal entity – a business owner. This could be, for example, a national subsidiary of a multi-national organization.

Especially in diversified service networks, there is often more than one entity with stakes in a service. For example, composite services are aggregations that comprise services from different providers. Each aggregated provider performs part of the composite service and hence becomes a stakeholder of the composite. This is due to the fact that it is the providers, who largely control the terms of engagement with aggregators concerning the re-use or re-purposing of their services. In other words, they have a certain influence on the composite service. Further examples of stakeholders include regulation bodies, like governments, that have the authority to prescribe aspects of service delivery, as well as third-party providers of delivery functions (e.g. billing or authentication) that are orchestrated with a service enabling the outsourcing of these functions. All these actors associated with parts of the delivery of a service are summarized under the term stakeholder.

A group of actors similar to stakeholders are so-called intermediaries. Like providers of delivery functions, they, too, provide value-added services. The difference is that they are involved in the provisioning and delivery of the service on a holistic level, i.e. the operations they perform encompass the entire service, not just a partial function.

Finally, the USDL also offers the possibility to relate services to taxonomies of user groups/profiles. This allows for the definition of general target consumers, as well as fine-grained customization of service offers (by limiting them to certain groups). It is left open who defines the taxonomies, though. Most probably there will be application or domain specific taxonomies, which have to be incorporated into an individual service description when the service is provisioned in the specific environment.

4.2.3.4 Functional Module

One of the most integral parts of every service description is to express what a service is able to do and how a consumer is able to use what the service provides. While the latter is well covered by existing technical service descriptions, at least for automated services, the former is less explored. In order to equally enable the description of human and automated services the Functional module captures both, conceptual functionality on one side and the technical realization that exposes this functionality on the other. It is important to distinguish between these two concepts, one being the subject of the service itself and the other being the service's interface. The reason is that a single service may be available, completely or in parts, over several interfaces. Interface in this context means a set of concrete technologies through which the service can be accessed. A simple example is an automated service that has a WSDL-based Web service interface and a REST interface. Even within one interface there might be different ways of access, e.g. by means of providing multiple access points that expect different sets of technical protocols to be used for communication. This concept is most prominently featured in WSDL with its ports and bindings.

Currently there are no standardized means to capture the functionality provided by a service. A common conceptualization, used for example in SoaML (OMG) or the SOA Reference Model (OASIS), is capability modeling. Oaks, Hofstede and Edmond discuss the concept of capabilities in [Oaks2003]. According to them, capabilities express the ability to perform a course of action that achieves a result. When viewed on a whole, this constitutes the functionality offered as *the* service. Actions themselves produce outcome, e.g., something is

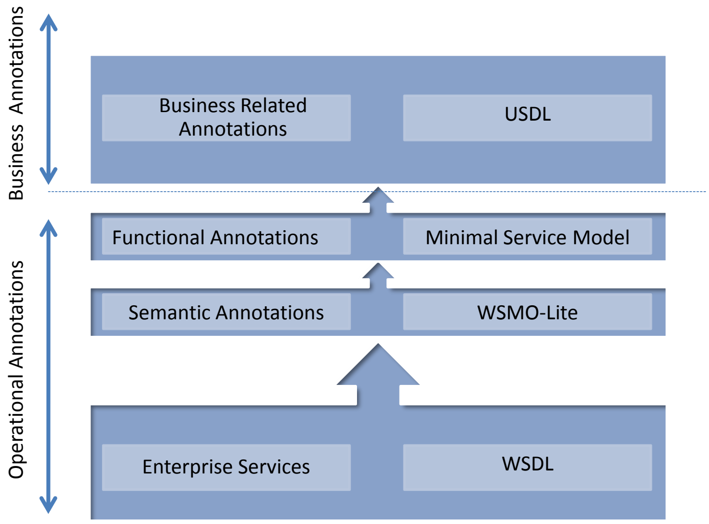

Figure 10: USDL and WSMO-Lite

created, transformed, or destroyed. Action also means that an actor is involved, who/which carries out the action. Usually, actors operate on one or more objects, consuming and producing some of them, while others are only affected. It is furthermore common that actors use resources as tools to perform an action. In some cases it is necessary to describe conditions that have to hold before an action can be started, as well as the effects that set in once the action is completed.

Exposing conceptual functionality of a service, similar concepts can be observed on technical interfaces of automated services. Such interfaces consist of operations that have typed input and output parameters. Additionally, faults/exceptions may be consumed or produced during an operation. Semantic service approaches further specify pre- and post-conditions of operations in a given expression language.

4.2.4 Describing Business Semantics in SOA4All

The service description in SOA4All is a lightweight set of semantic service descriptions in RDFS that can be used for annotations of various WSDL elements using the SAWSDL annotation mechanism. Using the WSMO-Lite ontology as the next evolutionary step after SAWSDL, filling the SAWSDL annotations with concrete semantic service descriptions and thus embodying the semantic layer of the Semantic Service Stack, enables SOA4All to identify the types and a simple vocabulary for semantic descriptions of services (a service ontology) as well as languages used to define these descriptions, to define an annotation mechanism for WSDL using this service ontology and to provide the bridge between WSDL, SAWSDL and (existing) domain-specific ontologies such as classification schemas, domain ontology models, etc. Being a lightweight semantic framework WSMO-Lite does not restrict the representation of the non-functional properties to a certain language or policy and leaves the issue of non-functional properties open. A complete service description including business semantics is still missing and can be achieved by applying the USDL.

An adequate service description can answer many questions of a service requester when searching for a service, like e.g. “When you encounter a service how do you determine how to request it? What is the identity of the service provider? Where and when is the service available? By what means do you access the service? What quality of service can you be guaranteed? What payment and settlement models are available? What rights as a service requestor do you have over the service? Where is the manifest that describes how the service is composed?” and can bridge the gap between business services and eServices. An enhanced representation of the service details, as provided by the USDL, can benefit the

SOA4All platform in different activities. The service discovery can be accelerated and become more accurate, through a detailed presentation of the service properties, services can be better compared, not only functional, but also according to the non-functional characteristics and thus an enhanced service substitution, enabling rational optimization and negotiation, is made possible. The dynamical and static service composition is empowered, as proven in this deliverable and advantages for the service management and evolution are also brought [OSullivan2006].

4.2.5 WSML Representation of the USDL

Because currently only an eCore presentation of the USDL is available, we modeled the published USDL concepts (some modules are still in development) as a WSML ontology. With USDL, we can add an additional semantic layer to the service descriptions of ES. By now ES have WSDL descriptions with functional properties represented via the minimal service model template (see Section 3.4). Information about service attributes beyond the functional properties can be added via the USDL (see Figure 10).

For the transformation of the USDL language into a WSML ontology we applied the rules given in Table 1. A small part of the transformed Core module is shown in Figure 11.

For linking between USDL and WSDL-files of the services an efficient solution is provided by USDL and adopted in our USDL-WSML-Ontology, namely the attribute concept in the `ElementDescription` concept, where we can save links to the artifacts of the WSDL service description. Using USDL we can as well describe human task services and provide thus detailed information of the provided human task like e.g. the included actions. Figure 12 gives an example.

eCore	WSML
Module	Ontology
Module dependency	Import Ontology
Interface/Abstract Class /Class	Concept (starts with capital letter)
Subclass	Subconcept (starts with capital letter)
Class attribute	Attribute (lower case)
Relations	Attribute(lower case, starts with typical relations expression e.g. has, is, etc)
Enum	Concept & Instances
Enum as attribute type	Relation between the class concept and the enum concept

Table *1*: USDL Transformation from eCore to WSML

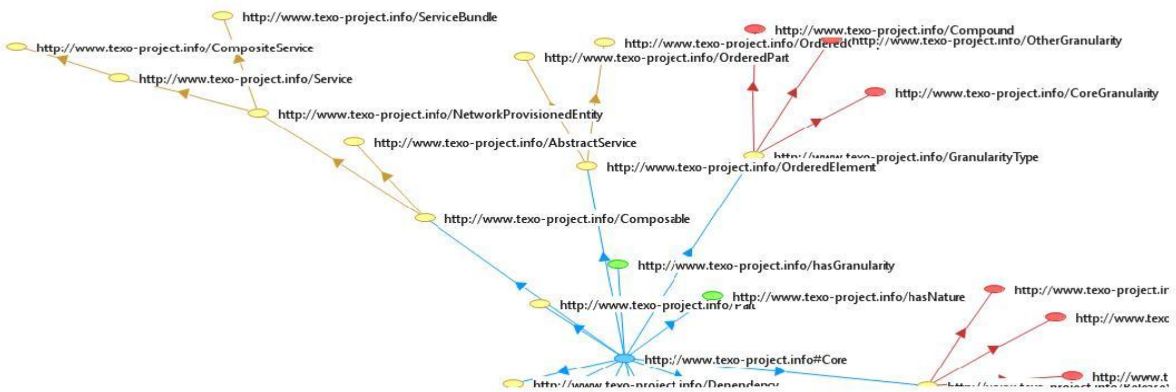

Figure 11: WSML Ontology for USDL Core Module

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"namespace {
_"http://www.texo-project.info/"
,
     sapesonto _"http://sap.com/xi/APPL/SE/Global/WSMO-Lite/ontology#",
     wsmostudio _"http://www.wsmostudio.org#" }

ontology USDL_AddressBasicDataByNameAndAddress
     nonFunctionalProperties
          wsmostudio#version hasValue "0.8.0"
     endNonFunctionalProperties

     importsOntology
          { _"http://www.texo-project.info/Foundation",
            _"http://www.texo-project.info/Functional",
            _"http://www.texo-project.info/Interaction",
            _"http://www.texo-project.info/Participants",
            _"http://www.texo-project.info/Pricing",
            _"http://www.texo-project.info#Core"}

….

instance InstanceAddressBasicDataElementDescription2 memberOf ElementDescription
     ElementName hasValue "EnterpriseServiceDescription"
     shortDescription hasValue "This is an SAP ERP service, which returns customer data
according to a query."
     ontologyConcept hasValue
"sapesonto#ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseService"
```

Figure 12: Sample Service Annotation

4.2.6 Applying Business Semantics in KPI-based Process Modeling

One of the biggest challenges for any kind of organization, private as well as public, is the alignment of the strategy with the operational activities. Most organizations are still struggling with the implementation of the organizational strategy. The term business performance management, which is often regarded as a further development of business intelligence, describes a "series of business processes and applications designed to optimize both the development and the execution of business strategy" [Frolick2006]. It consists mainly of two sequential tasks. The first one is the creation of strategic goals by the identification of specific objectives and KPIs that are of significant importance for the organization. Second, it supports the following management of the performance of those goals and the transformation of the indicators and the objectives in operational metrics, linked to performance actions. As part of the New Public Management, business performance management plays a significant role for public administrations. The BPM framework contains four steps (see Figure 13): strategize, plan, monitor and analyze, take corrective action [Frolick2006]. Applied to the public sector, the BPM framework can help organizations to improve their ongoing business operations and processes, which is a core step to an efficient eGovernment implementation [Becker2004].

During the "strategize" phase, a course of action is defined and key value drivers to attain the strategy and to generate metrics for performance measurement are identified. Finding timely and actionable measures that represent best the strategic value drivers is of significant importance for the success of the BPM framework. The identified measures used to mirror the organizational strategy are often called Key Performance Indicators (KPIs). Through strategy mapping the business value drivers and the corresponding KPIs can be derived from the organization`s vision. In the planning step a program of actions is developed, the concrete organizational outputs are formulated, which will carry out the organizational strategy. During this phase a detailed plan or budget is created that specifies how resources will be allocated. Business activities and processes having as target the achievement of the key performance indicators are planned and responsibilities are assigned to the business units. During the first two steps the strategy is defined and actions are planned, whereas the last two steps involve the strategy execution and the actual doing. The third phase of the BPM framework, monitor and analyze, enables constant monitoring of performance results. Using business intelligence and analytics the results are analyzed and reasons for performance gaps identified. Appropriate corrective actions are undertaken in the next step [Frolick2006]. We apply the BPM framework to a public sector organization and adopt it for service-based processes, which enables the organization to leverage the effects of the performance measurement by taking corrective actions before and after the process execution.

For public sector organizations strategy definition represents a considerable difficulty. The truth is that most strategy describing documents in public administrations contain a list of programs and initiatives, but not clearly defined outcomes and outputs, which the organization intends to realize [Kaplan2001]. Strategy is the core of each business performance activity and it is even more important in public sector organizations, whose motivation is not of financial nature but far more executing administration processes effectively and efficiently. Considering public administration`s strategy there are the mission, the vision and the overall concept to be taken into account [Stötzer2009]. The vision is a direction oriented ideal view of a central goal for the organization`s future. It describes a desirable future (most frequently in 3-5 years) positioning, to which all activities need to be aligned. The vision can be implicitly available in the organization or better, it is formulated in vision statements. The public value and functions of a public administration are part of the organization's mission. It contains the primary goal of an organization and gives reasons of its existence. In contrast to the vision, the mission must not be future oriented. It should describe the organizational intension, its objectives, core values, guiding principles und strategies. A mission statement is a written record of the mission and it focuses on what the institution really intends to do. Through the definition of the outcome of the organizational activities clear mission achievements are set. Because of its nature public administrations cannot define one measurable performance goal like it is the case in private companies; they rather have qualitative objectives like keeping the organization's progress or meeting the stakeholders' requirements [Stötzer2009].

An even bigger challenge than developing a winning strategy is the successful strategy implementation. Strategy statements can often lead to different interpretations by different employees for their everyday jobs. Thus, a strategy needs to be quantified and cascaded, in order to eliminate ambiguity and confusion about the organizational objectives [Kaplan2001].

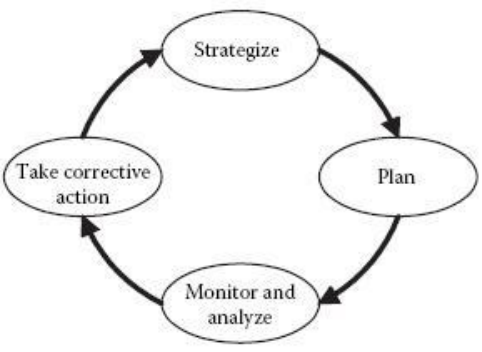

Figure 13: BPM Framework

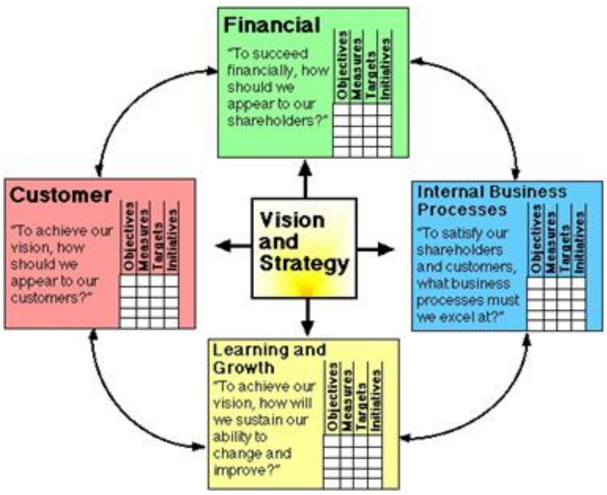

Figure 14: Balanced Scorecard

One method to transform the organizational mission into concrete performance measures is the Balanced Scorecard. The concept of the Balanced Scorecard was developed by Kaplan and Norton and represents a comprehensive framework that translates the corporation's strategic objectives into a coherent set of performance measures. In its raw version the Balanced Scorecards extends over four perspectives (see Figure 14): financial, customer, internal processes and learning and growth [Niven2008]. In contrast to traditional performance measurement approaches, that consider only financial measure, the Balanced Scorecard recommends the use of non-financial measures, in order to better understand how to sustain the financial performance in the future [Moore2003]:

1. In private sector organizations, the financial perspective dominates often the managers' interest, for it is an indicator for the long-term organizational success and strategy implementation. Typical financial measures are revenue growth, profitability and financial value and they are used to measure the financial success of the organization. In the non-profit sector, the goals are rather social than financial, it is not the sustained profitability, but the public value, which public sector organizations create. At the same time, they need to be as concerned about their financial viability (the ability to cover the costs of operating with revenues) as private organizations. Public administrations are also obliged to examine their costs without scarifying quality of their services. However, costs alone do not indicate to which degree the organization creates a public value [Cardoso2008].
2. The customer perspective emphasizes the strategy implementation from the customer's point of view. In order to choose measures for this perspective, two important questions have to be answered: Who are the target customers? And what is the value proposition in serving them? Customers are important for private companies, because buying the organizational products and service sustains the claim that the company is providing value to the society. In public administrations, delivering public services to their customers is the social justification for the existence and the activities of the organization [Moore2003].
3. The internal processes perspective identifies the key processes the firm has to excel in order to remain competitive. The internal process perspective contains indicators of operating performance (cost, quality, and cycle times) of critical processes that deliver value to customers and reduce operating expenses. Additionally, the internal perspective can include measures of innovation processes that create entirely new products and services [Kaplan2001].
4. The learning and growth perspective represents the state of sources as people and systems. Typical measures for this part of the Balanced Scorecard are employee motivation, retention, capabilities, and alignment, as well as information system capabilities [Kaplan2001].

In order to be applied to services provided by the public administrations, the balanced scorecard has to be adapted to the characteristics of the public sector domain and broken down to the administration process level, which is the main output of the activities of a public administration. Based on the organizational balanced scorecard, balanced scorecards for the single business units and for the business processes can be derived [Hermann2008]. From the Balanced Scorecard of the business unit, a Process Balanced Scorecard is derived, containing the following perspectives: process finance, process customers, process benefits and process potentials. For the different perspectives a set of measures is developed, some examples may be:

1. Process finance: process costs, process revenues, process yield
2. Process customers: customer satisfaction, customer commitment, process reliability

3. Process benefits: process temporal measures (throughput time, cycle time), process quality
4. Process potentials: employee's satisfaction, process improvements

The process scorecard provides a strategic conformed systematic process goals development by considering different perspectives of process performance [Hermann2008]. It can help public administrations to align the public services, offered by them, with their strategic goals and to monitor continuously the results of the delivered services. For the development of a process scorecard in the public administration, there are some specific characteristics of processes in municipal agencies which need to be considered [Becker2004], see *Table 2*.

Characteristic	**Description**
Immateriality of process objects	• Information processing functions predominate
Repeatability	• High number of cases • High amount of routine jobs
Linearity	• Many different process steps • Little parallel processing
Structurability, Stability, Consistency	• Based on legal regulations
Bilaterality	• High integration of customer (citizens/enterprises) • Many points of interaction
Decentrality	• In many cases highly fragmented, distributed working • Participation of many organization units inside and between authorities • Enormous flow of documents between individual stages

Table 2: Characteristics of Typical Administrative Procedures

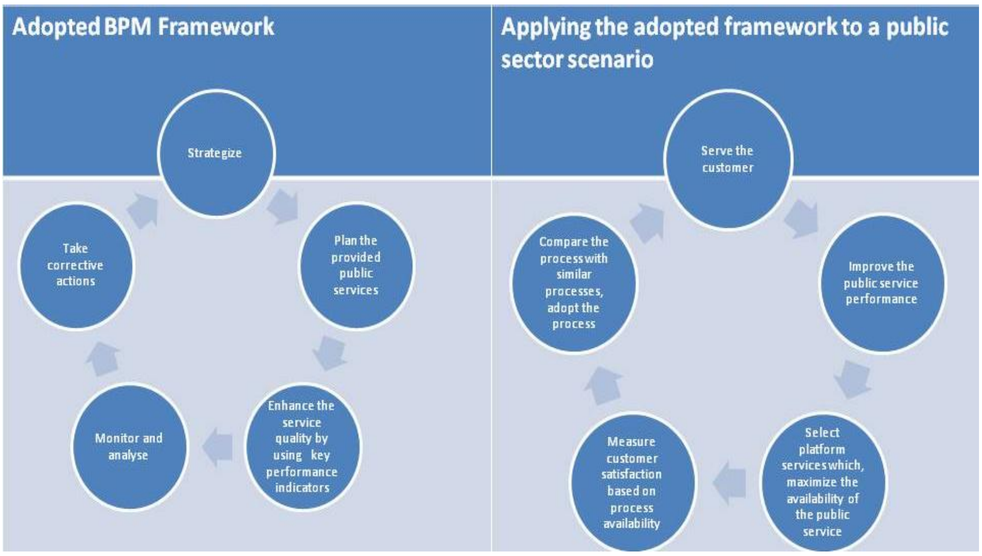

Figure 15: Process-Based Business Performance Management

Introducing a process scorecard for the public services of administrations can help public agencies to significantly improve the quality of their services. Considering the big picture, we can arrange the process scorecard in the organizational balanced scorecard by using the concept of a strategy map. Such a map shows the cause-and-effect chain between the strategic goals und represents their dependencies. The strategy map can be used for the control of the logical interdependence of the goals in and between the different scorecard perspectives. A strategy map contains only the goals that are relevant for an efficient public administration and provides a method for identifying the processes and the processes 'goals that best contribute to the targets reaching [Hermann2008].

The strategic steps mentioned above belong to the strategic process design, which has to be brought to the real processes by means of the operational process planning. The operational planning includes the following steps: selecting the performance indicators, setting of a measuring system with appropriate measures and planning the process goals. After the process execution, the actual process performance is measured and performance gaps are analyzed. For this purpose clear and quantifiable process goals need to be defined [Hermann2008].

Having the specific characteristics of public administration processes in mind and the rising demand of qualitative public services, SOA4All can not only enhance the design process of services by providing a process composer for dynamic process drawing and execution, but also accelerate and facilitate the operative process planning by offering public services, which best meet the strategic goals of the public administration office. Making use of a business user-oriented services selection for the public services can bridge the gap between the planned process and the provided business service. According to the different process goals, which may also differ for the different customers of the public administration, characteristic measures of the business service can be optimized by selecting the appropriate services for the single process activities. In order to compare Web services with each other, so that we can find out which one is in line with the desired public service, a semantic service description is needed, which is provided by the USDL. Using the SOA4All technologies we can enhance the Business Performance Framework in such a way to align the processes to the strategic and operational goals before and after their execution by providing an enhanced services selection and at the same time ex post process performance measurement. The adapted BPM framework and a concrete application to a public sector office are shown in Figure 15.

Binding the process goals to performance indicators enables the public administration to delineate the provided public services in such a way as to make them more efficient and effective. Process performance indicators build the basis for the Business Performance Framework. Critical parameter for the measurement of process effectiveness is the customer satisfaction, whereas for process efficiency the process time, the adherence to the delivery dates, the process quality and the process costs are of a significant importance. As mentioned above, processes goals depend on each other, which also means that the operational process metrics are related to each other and can be considered alone only rarely. For example, it can be expected that by decreasing the process time, increased process costs may arise [Hermann2008]. For this reason, we will provide the process modeler with a visualization of the process metrics values for the different services combinations as well as with optimized selected services for one or for more than one process metrics. Doing this will empower the business process creator to design the public services as planned in the strategic planning phase.

Having the characteristics of public administration processes and typical process goals like for example "Increase the process performance", a sample of a process scorecard for the online services of the City of Mannheim, Germany, was developed (see *Table 3*). Three major types of online services were identified and the perspectives of the process scorecard described above were adopted. Most services offered by public sector organizations are services provided to citizens or to private organizations (businesses, enterprises) or to other public sector organizations.

We broke down the operative goal samples to key performance indicators (optimization problems) and analyzed which service properties play a role in the measures computation (see *Table 4*).

Objectives	Customer perspective	Financial perspective	Internal Processes perspective	Process potentials
Citizens	Serve the customer • Increase service availability • Decrease service complexity	Meet budget restrictions • Increase benefit /cost • Decrease service costs • Increase number of self-provided services	Improve process quality • Decrease process time • Increase process compliance • Increase process security • Decrease number of responsible persons	Improve institutional constraints • Certificated services • Reduce energy consumption • Reduce paper
Businesses	Serve the customer • Increase service availability • Decrease service costs • Increase service confidentiality	Meet budget restrictions • Increase benefit/cost • Decrease service costs • Increase number of self-provided services	Improve process quality • Decrease process time • Increase process compliance • Increase process security • Decrease number of responsible persons	Improve institutional constraints • Certificated services • Reduce energy consumption • Reduce paper
Public Organizations	Serve the customer • Increase service availability • Increase service standardizati on • Increase service reliability	Meet budget restrictions • Increase benefit/cost • Decrease payment threats	Improve process quality • Decrease process time • Increase process compliance • Increase process security • Decrease number of responsible persons	Improve institutional constraints • Certificated services • Reduce energy consumption • Reduce paper • Use more public sector services

Table 3: Operative Goals for the Online Services of the City of Mannheim

	Service KPI	eService property	Function
1	Increase service availability	Availability	Maximize the product of the availability values
2	Decrease service complexity	Customer interactions	Minimize the number of services with interaction protocols
3	Decrease service costs	Cost	Minimize the overall sum of the price components
4	Increase service confidentiality	Confidentiality	Maximize the number of confidential services
5	Increase standard services	Classification: standard	Maximize the number of standard services, i.e. having the classification: standard
6	Increase service reliability	Reliability	Maximize the number of reliable services
7	Increase benefit /cost	Cost	Minimize service cost due to constant benefit
8	Increase number of self-provided services	Service Provider	Maximize the number of services with service provider the organization itself
9	Decrease payment threats	Payment methods	Minimize number of different payment methods
10	Decrease process time	Execution Time + Invocation time	Minimize overall execution time + invocation time
11	Increase security	Security	Maximize the number of secure services
12	Increase process compliance	Legal compliance	Maximize the number of legal compliant services
13	Decrease number of responsible persons	Human services	Minimize the number of service providers for human services
14	Reduce paper	Automotive services	Maximize the number of eServices
15	Reduce energy consumption	Classification: low energy consumption	Maximize the number of services with low energy consumption
16	Increase certification	Certificates	Maximize the number of certified services
17	Use more public sector services	Service Provider	Maximize the number of services having service provider a public sector organization

Table 4: Service KPIs for a Public Administration

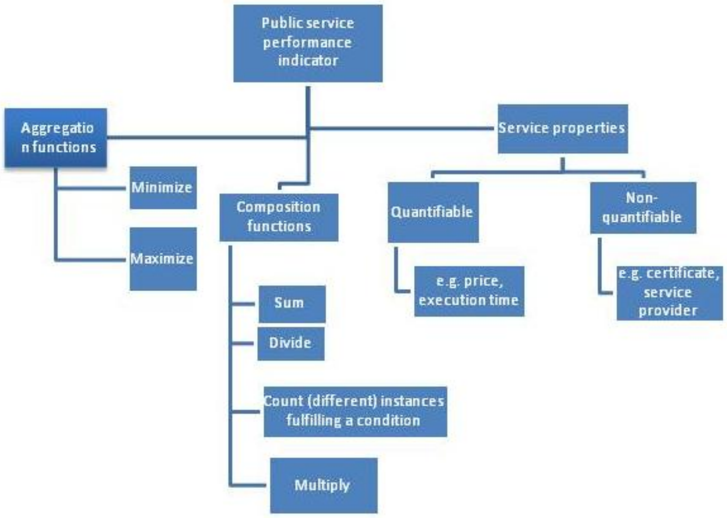

Figure 16: KPI Measurement

Based on the samples of performance indicators a simplified model of key performance measures for the online services of a public administration was developed and used for the implementation of the enhanced service selection (see Figure 16). The end-to-end measures of the public process are built by service the properties of the single eServices used. Most performance indicators aim either to maximize or to minimize some value, we call these functions aggregation functions. The value, which needs to be optimized through the service selection, is a composed value of service property values for the whole process, like e.g. the sum of the prices of the services, used for the process execution. Almost every performance indicator could be represented by the use of the composition functions and the service properties, described by the USDL. Some functions can be applied only to quantifiable service properties like e.g. execution time, price, and availability. Non-quantifiable properties like e.g. certificates, service providers can only be proven whether they fulfill a certain condition like e.g. services with ISO 9000 certificate or not and we can maximize/minimize the number of the services having the desired property value.

In SOA4All, the WP6 Optimizer will solve such service selection problems in business processes (see D6.4.1 and D6.4.2). Until a prototype is available, we have implemented a straightforward, graph-based algorithm described in Annex B to verify the general concept of KPI-based service selection.

4.2.7 Sample Scenario for the Public Sector

In order to show the additional benefit, provided by our approach, a public sector scenario for the Transportation Office of the city of Mannheim was developed.

According to a decision in the Transport Planning Office of the city of Mannheim, a reconstruction of the Richard Wagner Street is planned. Due to several complains about the street's quality in the recent years actions should be taken as soon as possible. However, before an official approval is given, a standard survey among the citizens living on this street should be executed. A survey layout is already available.

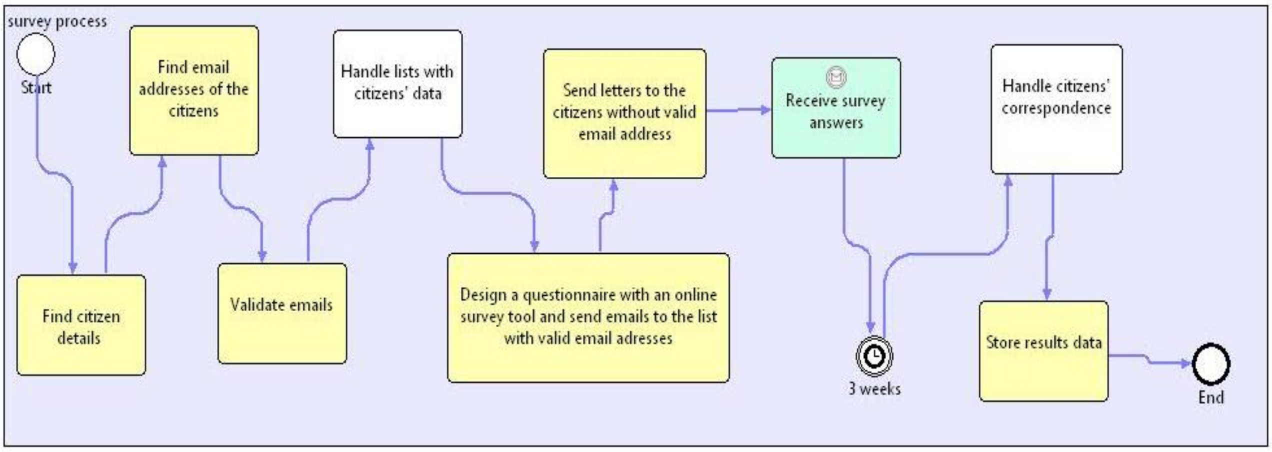

Figure 17: Survey Process at City of Mannheim

In order to standardize the survey process, the Transport Planning Office decides to design the process via the SOA4All platform. The actors in the survey process are:

- Civil servant as Service Composer at the Transport Planning Office
- Different eServices available via SOA4All
- Citizens of the city of Mannheim

A service composer at the TPO receives the task to model the survey process. After some discussions with colleagues, who have some experience with surveys, the process modeler designs the process via the SOA4All platform. The desired process contains the steps given in Figure 17:

1. The data of all adults living in the Richard Wagner Street are found. Every citizen of Mannheim has a record in the SAP CRM system of the city of Mannheim. The data is saved as an excel file.
2. The email addresses of these citizens are found. If someone in a registered household has an email address, the survey will be sent per email. The data is saved as an excel file.
3. The email addresses are validated, in order to assure that the emails will be sent correctly.
4. The TPO employee has to create two lists: one which contains all valid email addresses and the other one containing all names and addresses of the households without a valid email address.
5. The TPO employee has to create a survey form in an electronic form and in paper form.
6. The TPO employee sends the surveys in electronic form to the list of valid email addresses.
7. The TPO employee sends the survey letters per an online postal service.
8. The TPO employee receives the survey answers.
9. The results are stored as excel files.

After modeling the process with the SOA4All Composer and executing appropriate goal assignments, a functional service discovery is executed, which provides the Web services listed in *Table 5*.

Process Activity	**Web Services**
Find citizen details	ESILayerESAddressBasicDataByNameAndAnddress
Find email addresses of citizens	ESILayerESCustomerBasicDataByID
Validate emails	DOTSEmailValidation2; XWEBEMAILVALIDATION
Handle lists with citizens' data	CreateCustomerDataLists1; CreateCustomerDataLists2
Design a questionnaire	DesignQuestionnaire1; DesignQuestionnaire2; DesignQuestionnaire3
Send survey	SendPostalSurvey1; SendPostalSurvey2
Handle citizens' correspondence	HandleCorrespondence
Store results	StoreData1; StoreData2

Table 5: Result of Service Discovery

Each Service comes with a USDL-based description of Business Semantics as listed in Table 6.

Service property \ SOA4All-service	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2	XWEBEMAILVALIDATION	CreateCustomerDataLists1	CreateCustomerDataLists2	DesignQuestionnaire1	DesignQuestionnaire2	DesignQuestionnaire3	SendPostalSurvey1	SendPostalSurvey2	HandleCorrespondence	StoreData1	StoreData2
availability	99.9	99.9	**99'995**	98	90	90	99	95	90	95	95	90	99	99.9
interaction protocols	none	none	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no
currency	€	€	$	$	€	€	€	$	€	€	€	€	$	$
price component	UsageBasedFee	UsageBasedFee	UsageBasedFee	UsageBasedFee	UsageBasedFee	UsageBasedFee	OneSurveyPacket	FreeSurvey	UsageBasedFee	UsageBasedFeeStandardLetter	UsageBasedFeeStandardLetter	UsageBasedFee	UsageBasedFee	UsageBasedFee
price level	AbsolutPriceLevel	AbsolutPriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel	AbsolutePriceLevel
price metric	Invocation	Invocation	Invocation	Invocation	Invocation	Invocation	Questions	Questions	Invocation	Letter	Letter	Invocation	GB	2
tierHigherBound			500	10000			30	10					2	2
amount	0.5	0.5	0	5	14.5	10.5	198	0	28.75	0.55	0.55	28.5	0	0.6
price component			One-timeSetupFee	PerHitFee			UsageBasedFee	AdvancedUserFee		UsageBasedFeeHumanResource	UsageBasedFeeHumanResource		UsageBasedFeePremium	UsageBasedFeePremium
price level			AbsolutePriceLevel	AbsolutePriceLevel			AbsolutePriceLevel	AbsolutePriceLevel		AbsolutePriceLevel	AbsolutePriceLevel		AbsolutePriceLevel	AbsolutePriceLevel
price metric				Invocation			Invocation	Responses		Invocation	Invocation		GB	GB
tierLowerBound				10000										
tierHigherBound								1000					50	50
amount			0	0.01			14.5	19.95		28.75	28.75		9.99	15
price component								UsageBasedFeeHumanResource			UsageBasedFeeNumberSentLetters			
price level								AbsolutePriceLevel			AbsolutePriceLevel			
price metric								Invocation			Number of sent letters			
tierLowerBound											50			
tierHigherBound											99			
amount								14.5			0.64			
confidentiality	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes
classifications	Standard	Standard	Commercial	Commercial	Standard	Standard	Standard	Non-standard	Standard	Non-standard	Non-standard	Standard	None	None
reliability	98.2	98.2	99.9	98	78	78	98	97	78	91	91	78	97	99.9
service provider	SAP	SAP	ServiceObjects	XWebServices	TPO_Mannheim / Regular public servant	TPO_Mannheim / Public trainee	2Ask ; TPO_Mannheim/Regular public servant	SurveyMonkey ; TPO_Mannheim/Regular public servant	TPO_Mannheim / Regular public servant	Deutsche Post /TPO_Mannheim/Regular public servant	PixelLetter/TPO_Mannheim/Regular public servant	TPO_Mannheim / Regular public servant	Dropbox	Amazon
payment methods														

provider	SAP	SAP	PayPall	XWebServices	TPO_Online_Payment	TPO_Online_Payment	Visa;Master; TPO_Online_Payment	Visa;Master; PayPall;TPO_Online_Payment	TPO_Online_Payment	Visa;Master; PayPall;TPO_Online_Payment	Visa;Master; PayPall;TPO_Online_Payment	TPO_Online_Payment	Visa	Visa
payment instruments	Online	Online	Online	Cashiers Check, Money Order, Credit Card	Online	Online	Credit card; Online	Credit card; Online	Online	Credit card; Online	Credit card; Online	Online	Credit card; Online	Credit card; Online
payment modality	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit
execution time/ invocation time	0.5/ 5	0.5/ 5	0,4/ 5	0.5 / 5	1800/ 43200	3600/ 43200	1800/ 43200	1800/ 43200	3600/ 43200	172800/43200	172800/43200	3600/ 43200	0,1/4	0,1/4,5
security	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
legal compliance	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
service nature	automated	automated	automated	automated	human	human	semi-automated	semi-automated	human	semi-automated	semi-automated	human	automated	automated
classifications	Low energy consumption	Low energy consumption	None	None	High-energy consumption	High-energy consumption	Medium-energy consumption	Medium-energy consumption	High-energy consumption	Low-energy consumption	Medium-energy consumption	High-energy consumption	Low-energy consumption	Low-energy consumption
certifications	Netweaver 7.0	Netweaver 7.0	None	None	Certificated	None	None	None	Certificated	Certificated	Certificated	Certificated	Certificated	Certificated
service provider classification	Non-public	Non-public	Non-public	Non-public	Public	Public	Non-public	Non-public	Public	Public	Non-public	Public	Non-public	Non-public

Table 6: USDL-based Business Semantics for selected Web Services

The selection of the different performance indicators will provide different services combinations as listed in Table 7. For example, if the business user decides that it is important to decrease the payment treats of the process, all combinations of different payment methods are searched through and the service combination, for which the different number of payment instruments is minimal, is selected. For this case, an exhaustive search is needed in order to find the optimal solution, whereas for the minimisation of the process costs, a shortest path algorithm suffices.

Service KPI	Find citizen details	Find email addresses of citizens	Validate emails	Handle lists with citizens' data	Design a questionnaire	Send survey	Handle citizens' correspondence	Store results
1 Increase service availability	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData2
2 Decrease service complexity	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1; DesignQuestionnaire2; DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
3 Decrease service costs	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	XWEBEMAILVALIDATION	CreateCustomerDataLists2	DesignQuestionnaire2	SendPostalSurvey1	HandleCorrespondence	StoreData1
4 Increase service confidentiality	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1; DesignQuestionnaire2; DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData2
5 Increase standard services	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
6 Increase service reliability	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData2
7 Increase benefit /cost	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	XWEBEMAILVALIDATION	CreateCustomerDataLists2	DesignQuestionnaire2	SendPostalSurvey1	HandleCorrespondence	StoreData1
8 Increase number of self-provided services	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
9 Decrease payment threats	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
10 Decrease process time	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2	CreateCustomerDataLists1	DesignQuestionnaire1; DesignQuestionnaire2;	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1
11 Increase security	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1; DesignQuestionnaire2; DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
12 Increase process compliance	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1; DesignQuestionnaire2; DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
13 Decrease number of responsible persons	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1	DesignQuestionnaire1; DesignQuestionnaire2; DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
14 Reduce paper	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1; DesignQuestionnaire2;	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
15 Reduce energy consumption	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1; DesignQuestionnaire2;	SendPostalSurvey1;	HandleCorrespondence	StoreData1; StoreData2
16 Increase certification	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1	DesignQuestionnaire3	SendPostalSurvey1; SendPostalSurvey2	HandleCorrespondence	StoreData1; StoreData2
17 Use more public sector services	ESILayerESAddressBasicDataByNameAndAddress	ESILayerESCustomerBasicDataByID	DOTSEmailValidation2; XWEBEMAILVALIDATION	CreateCustomerDataLists1; CreateCustomerDataLists2	DesignQuestionnaire1; DesignQuestionnaire2; DesignQuestionnaire3	SendPostalSurvey1	HandleCorrespondence	StoreData1; StoreData2

Table 7: Selected Services based on KPI

5. Goals (Service Templates)

Despite the fact that a project-wide *goal model for service* and *goal specification* is still pending, abstract goal specification from a lightweight and pragmatic perspective is emerging as on-going work in WP3. *Service templates* [Lambert2010] represent a novel approach to simplify the stateful WSMO based [Roman2005] service discovery and matchmaking which relies on a goal driven approach to match stateful description of user capability desire in goal with stateful capability functional description of service interface. Expressive reasoning is support that does not always guarantee decidability if rich and expressive condition and effects axioms are involved. Furthermore, despite its W3C submission status, it remains unclear whether business users should be expected to write logical axioms to express their goal.

The minimal service model (MSM) [Lambert2010] brings in the new service templates as a declarative notation and specification mechanism to allow indirect, simplified goal specification; focusing on the essential elements within service request or response instances in RDF and simultaneously catering for scalability. Instead of denoting condition constraints using expressive logical axioms directly with all its extra challenges required in expertise and myriad of overhead involved in the process, service templates are catered for easier formulation of goal that is linked declaratively to request/response instances. They contribute to a more lightweight approach to annotate extra semantic information into an on-going work on goal model. They allow subsequent dynamic generation of SPARQL queries using work-in-progress algorithms wrt a given service template in order to query for services annotated in WSMO-Lite/MSM in an RDF *knowledge base*.

5.1 Service Templates Mechanism

The service template approach is centered to achieve scalability by easing goal formulation to a certain non-technical target group, especially laymen, without further assistance. It represents a more abstract approach that allows goal specification based on service request in parallel to known service descriptions (MSM/WSMO-Lite or MSM/MicroWSMO service ontologies). An important design decision is about a straightforward mapping of service templates into SPARQL queries. The rationale behind this decision lies in the fact that WSMO-Lite/MicroWSMO service annotations are generally available in the SOA4All Semantic Space which serves as a huge *knowledge base* (KB). A simple definition of the set of service template related concepts are shown in the following service template ontology listing.

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix msm: <http://cms-wg.sti2.org/ns/minimal-service-model#>.
@prefix st: <http://cms-wg.sti2.org/ns/service-template#>.
 st:ServiceTemplate a rdfs:Class.
 st:hasFunctionalClassification a rdf:Property;
     rdfs:domain msm:Service;
     rdfs:range rdfs:Resource.
 st:hasInput a rdf:Property;
     rdfs:domain msm:Service;
     rdfs:range rdfs:Resource.
```

```
st:hasOutput a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:hasPreference a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:hasRequirement a rdf:Property;
    rdfs:domain msm:Service;
    rdfs:range rdfs:Resource.
st:value a rdf:Property;
    rdfs:domain rdfs:Resource;
    rdfs:range rdfs:Datatype.
```

In the above listing, three properties are most relevant for a simplified service discovery, matchmaking approach based on service templates:

- `st:hasFunctionalCategory`
- `st:hasInput`
- `st:hasOutput`

The `st:hasFunctionalCategory` RDF-property is obviously a link provider to include our previously described functional classification structure to further assist service discovery over SPARQL and simplify matchmaking because a well-structured classification hierarchy helps graph-pattern matching in SPARQL selection more efficiently by following a well-defined guiding classification structure and allowing search space pruning to a large extend during SPARQL query execution.

The `st:hasInput` property is the central linkage piece for relating the service template to the input message instance and allowing declaration of a list of unnamed blank RDF nodes to 'reach out' in the search space of the RDF-graph containing constituent instances in the input message that might be a match to some search criteria. In this regard input message within the service templates resembles a substitute for the definition of more heavyweight precondition axioms that are typical in a stateful service matchmaking mechanism.

In a similar fashion, `st:hasOutput` provides the slot for linking in an instance of the output message which can be viewed, as described previously, as a lightweight substitute for postcondition or effect axioms. What is essential is the notion of dynamic SPARQL query generation in this mechanism since service templates only provide a way for abstract declarations without having a means to actually query for service matches. A similar generative approach [Sbodio2007] has been attempted based on the OWL-S service ontology and information model.

As momentarily semi-automated SPARQL generation based on service templates are still on-going work in WP3, it remains to be the target of the next practical proof-of-concept for the service template approach to excel.

Regarding content of the service template for our example ES, we have provided the content to show exemplarily how this mechanism is supposed to work in the following listing.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix msm: <http://cms-wg.sti2.org/ns/minimal-service-model#>.
@prefix st: <http://cms-wg.sti2.org/ns/service-template#>.
@prefix saperpsharedonto: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:shared_onto#>.
@prefix refwsmoliteonto: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_wsmolite#>.
@prefix refmsmonto: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_msm#>.
@prefix tns: <urn:un:eu:soa4all:transform:interoperation:standard:www.sap.com:erpsv_servicetemplate #>.
################################
# service template declaration
################################
tns:ServiceTemplate_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceRequest
    a st:ServiceTemplate;
         st:hasFunctionalClassification
refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceFunctionalClassificationInstance;
         st:hasInput [ rdf:type refwsmoliteonto:CustomerERPAddressBasicDataByNameAndAddressQuery_syncElement ];
         st:hasInput [ rdf:type msm:Message ];
         st:hasInput [ rdf:type refwsmoliteonto:CustomerERPAddressBasicDataByNameAndAddressQueryMessage_sync ;
                                    rdf:type saperpsharedonto:CustomerSelectionByNameAndAddress ;
                                    rdf:type saperpsharedonto:ProcessingConditions ;
                                    rdf:type saperpsharedonto:FirstLineName ;
                                    rdf:type saperpsharedonto:SecondLineName ;
                                    saperpsharedonto:hasFirstLineName tns:FirstLineNameInstance ;
                                    st:value "Martin"^^xsd:string ;
                                    saperpsharedonto:hasSecondLineName tns:SecondLineNameInstance ;
                                    st:value "Mustermann"^^xsd:string ;
                                    rdf:type saperpsharedonto:TaxJurisdictionCode ;
                                    st:value "0414316731"^^xsd:nonNegativeInteger ;
                                    saperpsharedonto:hasCountryCodeElement tns:CountryCodeElementInstance ;
                                    saperpsharedonto:hasRegionCodeElement tns:RegionCodeElementInstance ;
                                    saperpsharedonto:hasCityName tns:CityNameInstance ;
                                    saperpsharedonto:hasDistrictName tns:DistrictNameInstance ;
                                    rdf:type saperpsharedonto:CountryCode ;
                                    st:value "US"^^xsd:string ;
                                    rdf:type saperpsharedonto:RegionCode ;
                                    st:value "WA"^^xsd:string ;
                                    rdf:type saperpsharedonto:CityName ;
                                    st:value "Tacoma"^^xsd:string ;
                                    rdf:type saperpsharedonto:DistrictName ;
                                    st:value "Washington"^^xsd:string ] ;
         st:matchDegree st:ExactMatch .
```

Our initial attempt to declare the service template is based on real-world service request message and valid request message as well as meaningful parameter values of the example ES. Declaration of the service template is highlighted in the above listing. Furthermore we see the application of our described functional classification of the ES in the listing to ease search and matching. The `st:hasInput` RDF-property serves to bind three lists of RDF *blank nodes* using the square bracket blank node shorthand in order to declaratively identify parameter instance in the semantic space and allow subsequent generation of meaningful SPARQL query parameter or SPARQL filter based on the provided parameter values. Noteworthy is that these parameter values have be adopted from real-world values in the request body.

It is worth mentioning the `st:matchDegree` RDF-property and its range taking an instance of `st:ExactMatch`. Service template matching degree will be discussed in the next section with regard to our example service template. Here the `st:ExactMatch` instance declaratively constraints the allowed *type of match* for the input message. Consequently generated SPARQL query of this template should have exact match semantics when the SPARQL is evaluated by a SPARQL engine against the known knowledge base.

5.2 Querying and Matching

Following our example above we have provided valid SPARQL queries, among others one that caters a simple exact match semantics with results bound to the four variables following the SELECT keyword. It is shown in the following listing.

```
###################################################################
# simple query for discovery based on functional classification
###################################################################
SELECT ?service ?operation ?serviceOntologyConceptRefURI ?serviceDeploymentURI
WHERE {
    ?service rdf:type msm:Service ;
         msm:hasOperation ?operation ;
         sawsdl:modelReference ?serviceReferenceURI ;
         msm:hasFunctionalClassification ?functionalClassification .
    ?service rdf:type wsl:Service ;
         rdfs:isDefinedBy ?serviceDeploymentURI .
    ?functionalClassification rdf:type sapesfc:BusinessDomainFunctionalClassification ;
         sapesfc:hasBusinessDomainOperationalClassification
sapesfci:BusinessDomainFunctionalClassificationSearch ;
         sapesfc:hasBusinessDomainDesignationClassification
sapesfci:BusinessDomainFunctionalClassificationCRM ;
         sapesfc:hasBusinessDomainExtensibleFunctionalClassification
sapesfci:BusinessDomainFunctionalClassificationPublicSector .
    ?operation msm:hasInputMessage ?input ;
         msm:hasOutputMessage ?output .
}
```

As shown here it becomes apparent from the shown namespace prefixes that the minimal service model and the underlying WSMO-Lite service ontologies must be referenced in the Semantic Space during query execution. Obviously such dependency and characteristics are aligned with the intension of the minimal service model and service templates as a means to support service discovery and matchmaking based on lightweight service annotations.

If the SPARQL query can execute to bind instances from any matched RDF sub-graphs, we can obtain concept instances for the semantic ES, its operation, its dependency on other service ontology and a URI to actually invoke the service. The given example above can be found in the project repository in same location as described and in the file

query_sap_esi_layer_es_address_basic_data_by_name_and_address.sparql.

The matching mechanism of service templates resembles the well-known service capability matching types [Paolucci2002], such as exact match, plug-in match, subsumes match, etc. This information is reflected in the service template model in the following listing.

```
#############################
# service template matching
#############################
st:MatchingResults a rdfs:Class.
st:MatchDegree a rdfs:Class.
st:ExactMatch a st:MatchDegree.
st:PluginMatch a st:MatchDegree.
st:SubsumesMatch a st:MatchDegree.
st:matches a rdf:Property;
    rdfs:domain st:ServiceTemplate;
    rdfs:range msm:Operation.
st:matchDegree a rdf:Property;
    rdfs:domain st:ServiceTemplate;
    rdfs:range st:MatchDegree.
```

Taking the concept of st:MatchDegree into account, [Lambert2010] has specified three instances of matches as shown above. We have applied the notion of subsumes-matching to a looser service template declaration to include subsumes matching in the following listing.

```
tns:ServiceTemplate_ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseEnterpriseServiceResponse
    a st:ServiceTemplate;
        st:hasFunctionalClassification
refwsmoliteonto:ESILayerESCustomerERPAddressBasicDataByNameAndAddressQueryResponseServiceFunctionalClassificationInstance;
        st:hasOutput [ rdf:type refwsmoliteonto:CustomerERPAddressBasicDataByNameAndAddressResponse_syncElement ];
        st:hasOutput [ rdf:type msm:Message ];
        st:hasOutput [ rdf:type saperpsharedonto:FirstLineName ;
                            rdf:type saperpsharedonto:SecondLineName ;
                            saperpsharedonto:hasFirstLineName tns:FirstLineNameInstance ;
                            st:value "Martin"^^xsd:string ;
                            saperpsharedonto:hasSecondLineName tns:SecondLineNameInstance ;
                            st:value "Mustermann"^^xsd:string ;
                             rdf:type saperpsharedonto:CountryCode ;
                            st:value "US"^^xsd:string ;
                            rdf:type saperpsharedonto:RegionCode ;
                            st:value "WA"^^xsd:string ;
                            rdf:type saperpsharedonto:CityName ;
                            st:value "Tacoma"^^xsd:string ;
                            rdf:type saperpsharedonto:DistrictName ;
```

```
                                      st:value "Washington"^^xsd:string ] ;
        st:matchDegree st:ExactMatch ;
        st:matchDegree st:SubsumesMatch .
```

It remains again up to the upcoming service templates to SPARQL generation algorithms in WP3 to lift up to the upper echelon and technical demand to generate SPARQL queries that actually realizes a subsumes-match, for instance, by insertion of appropriate SPARQL *filter* into the generated queries in order to realize the semantics of a subsumes match. The rationale lies supposedly again in taking advantage of specifying a syntax agnostic generative model.

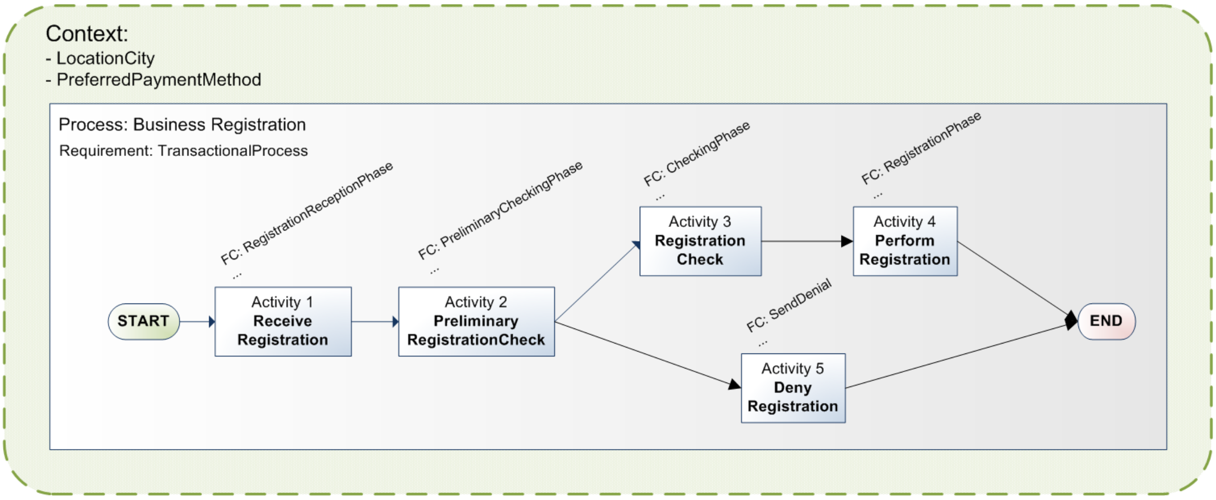

Figure 18: Business Registration Process with Context–dependent Execution

6. Context-Aware Process Models

While the previous Chapter investigated the application of the novel Services Template from the Minimum Service Model within the WP7 use case, the SOA4All Process Modeling and Execution by WP6 currently relies on the traditional, WSMO-based goal definition (see D6.3.2 and D6.4.2) and may be extended in the remainder of the project. In this Chapter, we therefore describe a working example for a business process that contains such a goal (also see D7.3), which is contextualized. The process is modeled as LPML process (see D6.3.1 and D6.3.2) and therefore support for dynamic, context-based discovery of services that fulfill the defined goal at design time or runtime of the process.

6.1 A Context Ontology for WP7

Contextual information can be used, e.g., to filter the potentially large number of Web services such that either the user can find a fitting service much faster or that a fitting service can be selected automatically during process composition (see D6.4.2). For the purpose of discovering Web services that match the given activities, a contextual classification for the eGovernment domain has been created in WP7. This context ontology can be integrated with a general SOA4All context ontology once that has been defined by WP3. The context ontology is used to annotate individual process activities as well as entire processes, e.g., directly within the SOA4All Composer. In Figure 18, a simplified business registration process is depicted. In this example, the context is depicted outside the scope of the process since it is influencing both the process and the individual activities.

Here we consider to contextual dimensions, namely the citizen context and the organization context:

(1) The citizen context is inferred from the user's profile and includes:

- User-ID
- Name
- Location
    - Country
    - City

- Language
- Preferred method of payment

(2) From the organization point of view, the context comprises:

- Organization
- Department
- Preferred method of payment
- Preferred acknowledgment method
- Preferred reception method

The activities using this contextual information specializes the process to the organizational aspects. Each organization department might have a different policy with regards to the method of payment. In that case, the process will have to be adapted at design time to the specific payment method in use by that department. In the same way, depending on the organization involved, the submission or reception of the form might be mandatory to be done in paper form or by electronic means. At execution time, when taking the user location into account, a process may resolve to send the registration form to the user own council, making use of the user's context.

The context information is built on top of the service description and extends it by adding the `hasSensitivityTo` property to the service. Through a relation between the service and the context, querying services and goals over the context become possible. Next follows an excerpt from a domain-based context model that takes into account the city and preferred payment method of a user.

```
concept PaymentMethod subConceptOf Payment
concept Card subConceptOf PaymentMethod
instance Visa  memberOf Card
concept City
instance Madrid memberOf City
concept PersonalProfile
concept Person subConceptOf PersonalProfile
  hasCitizenship ofType City
  hasPreferredPaymentMethod ofType PaymentMethod
instance Pascale memberOf Person
hasCitizenship hasValue Madrid
hasPreferredPaymentMethod hasValue Visa
concept Service
   hasSensitivityTo ofType Dimension
concept Dimension
   hasDimensionValue impliesType DimensionValue
concept DimensionValue
instance MadridCityDimension memberOf Dimension
hasDimensionValue hasValue Madrid
hasAssociatedSlot hasValue hasCitizenship
instance Pascale memberOf Person
hasCitizenship hasValue Madrid
hasPreferredPaymentMethod hasValue Visa
instance Service1 memberOf msm#Service
   hasSensitivityTo hasValue MadridCityDimension
   hasFunctionalClassification hasValue CitizenRegistration
relation select (ofType Person, ofType Service)
relation citizenshipContext (ofType Person, ofType Service) subRelationOf select
```

Once the context is modeled, it is possible to query for services specific for a given context:

```
?x[
       msm#hasFunctionalClassification hasValue
ego#PaymentOrderForBusinessRegistration,
       msm#hasOperation hasValue ?operation
] memberOf msm#Service
and
?operation [
       msm#hasInputMessage hasValue css#citizenCRMforkParameter,
       msm#hasOutputMessage hasValue css#salesOrder
] memberOf msm#Operation
and wp7c#preferredPaymentContext (?x, ?y)
and wp7c#citizenshipContext (?x, ?y)
and ?y = wp7c#Pascale
```

6.2 A Context-Aware LPML Process

LPML models are using the concept of a goal activity to abstract from concrete services and describe what is expected from an activity instead (see D6.3.1 and D6.3.2). Goals are represented by a set of semantic annotations attached to a particular activity that describe its functional and non-functional characteristics. When a business user is modeling a process in the SOA4All Composer, he may use predefined semantic annotations to annotate activities and define a goal. Later, that information is used by the WP6 Design Time Composer that is searching for a matching service or process template that satisfies the goal (see D6.4.2). In that sense, goals are providing a higher-level layer of abstraction, while process composition is a process of making process more specific, going from abstract level into concrete (and if possible) executable process.

An LPML goal has the following elements: `FunctionalClassification`, `Input`, `Output`, `Requirement`, and `Constraint`. Additionally, goals are being resolved in some "context". Goal context is the following:

- Global context: the context of a process as a whole (that is specified by global process annotations) that is used for composition purposes.
- User specific context: the context of a current user performing process (that what is called Context in the meaning of SOA4All) - information that can influence service selection, based on user profile, etc. (see D3.2.1).

Going back to the example from above, the Business Registration Process will be resolved by the WP6 Design Time Composer by performing a template expansion and by replacing some activities with appropriate, specialised templates. Those templates contain further goals that are more specific and provide more details that drive the composition process. In Figure 19, one activity is depicted in detail, showing all attached semantic annotations.

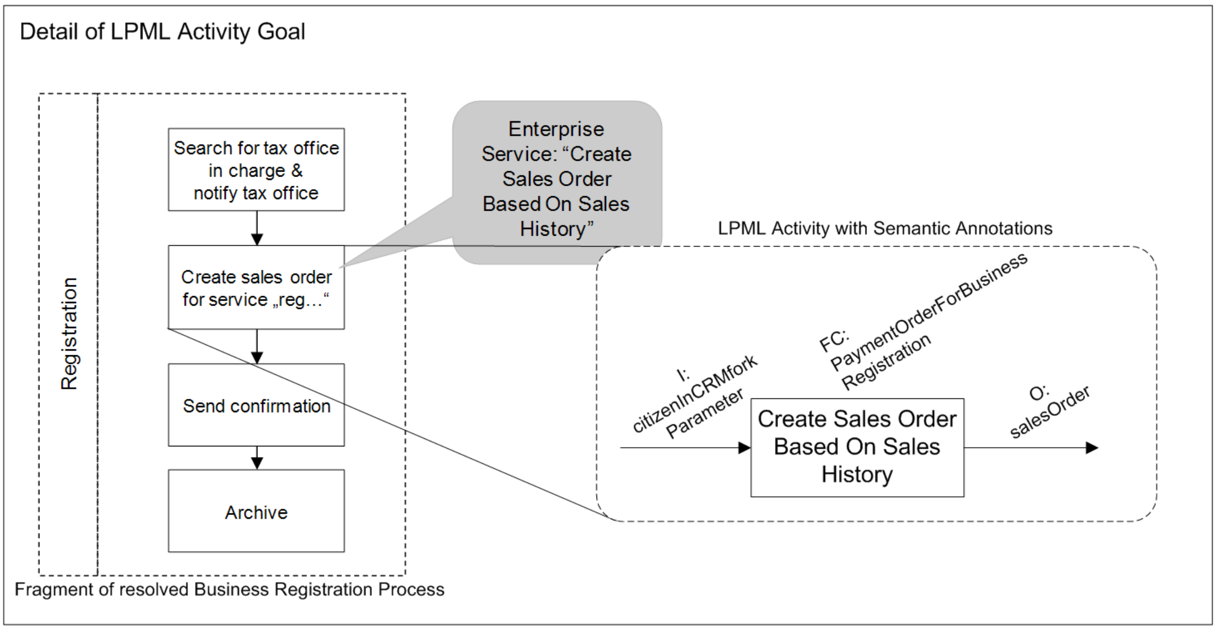

Figure 19: LPML Activity Goal

The purpose of the activity "Create Sales Order Based On Sales History" is to perform the registration payment process. The goal of this activity is the following (namespaces and unnecessary information were removed for brevity):

LPML Goal:

- `FunctionalClassification: RegistrationPhase`
- `FunctionalClassification: CitizenRegistration`
- `Input: citizenInCRMforkParameter`
- `Output: salesOrder`

In this example, based on that goal the WP5 Service Discovery is used to provide services compatible with following annotations. A query to retrieve compatible service is given below:

```
======msm
?service [
     hasFunctionalClassification hasValue RegistrationPhase,
     hasFunctionalClassification hasValue CitizenRegistration,
     hasOperation hasValue ?operation
] memberOf Service
and
?operation [
     hasInputMessage hasValue citizenInCRMforkParameter,
     hasOutputMessage hasValue salesOrder
] memberOf Operation .


======= WSMO lite
?service [
     hasFunctionalClassification hasValue RegistrationPhase,
     hasFunctionalClassification hasValue CitizenRegistration,
     hasCondition hasValue citizenInCRMforkParameter,
     hasEffect hasValue salesOrder
] memberOf Service .
```

The context of a process influences the selection of services during goal matching. In this example, we the following user-specific context is used: location of user (city) and preferred payment method. A sample query is given below:

```
?service [
     hasFunctionalClassification hasValue RegistrationPhase,
     hasFunctionalClassification hasValue CitizenRegistration,
     hasOperation hasValue ?operation
] memberOf Service
and
?operation [
     hasInputMessage hasValue citizenInCRMforkParameter,
     hasOutputMessage hasValue salesOrder
] memberOf Operation
and
citizenshipContext (?service, ?citizen)
and
?citizen = Pascal .



===wsmo lite
?service [
     hasFunctionalClassification hasValue RegistrationPhase,
     hasFunctionalClassification hasValue CitizenRegistration,
     hasCondition hasValue citizenInCRMforkParameter,
     hasEffect hasValue salesOrder
] memberOf Service

and
citizenshipContext (?service, ?citizen)
and
?citizen = Pascal .
```

7. Conclusions

WP7 implements an open service delivery platform that allows civil servants to handle typical administrative procedures (such as an application for registering a new business). More specifically, using the Web-based tools of the SOA4All Studio, civil servants can search, model, annotate, modify, share, analyze, and execute administrative procedures in the form of lightweight processes. These processes may be composed of Enterprise Services (hosted by SAP), public Web services (hosted by 3rd party service providers), and human activities (to be executed by end users).

The service delivery platform can largely be implemented by leveraging the functional components provided by the technical work packages of SOA4All. But in order to meet all requirements of the use case scenario, some customizations and extensions are developed within WP7. Thus, in addition to investigating the public sector business case and to providing a basis for the technical validation and end user evaluation of the project results, the key technical contribution of WP7 to SOA4All is an adaptation and integration layer so that SAP Enterprise Services (ES) can be consumed in SOA4All. As of now, SAP provides its ES with WSDL-based Web service interfaces only. In this Deliverable, we have therefore defined semantics on several levels that are required to fully integrate SAP ES into the advanced semantic Web platform of SOA4All: (1) we defined a lightweight domain ontology for SAP Enterprise Applications. This domain ontology captures the distinct characteristics of SAP Enterprise Service (ES) and serves as a basic building block upon which more specific ontologies can be created. (2) Next, we described the functional Web service annotations using the WSMO-Lite service model to annotate an example SAP ES reusing the defined domain ontology. (3) Then we presented a novel approach to express pre-condition and post-condition axioms, which can be transformed to SPARQL queries for service discovery. (4) We have extended the ES annotations with non-functional properties and (5) introduced a novel approach to describe business semantics based on the Unified Service Description Language (USDL). (6) In a next step, we used these business semantics for selecting services based on the specific, context-dependent business requirements of a SOA4All user during service invocation and service composition (process modeling). (7) Furthermore, we discussed the Service Templates approach as a novel approach to simplify service discovery and matchmaking with a simplified goal specification focusing on the essential elements within service request or response instances in RDF. (8) Finally, we presented a working example of a contextualized business process that contains a context-sensitive goal as process activity. (9) In Annex A, we also survey existing ontologies that have been or are currently defined by related European and national projects and analyze to what extent they can be reused in the scope of this use case.

All artifacts described in this Deliverable (ontologies, annotated Web services, semantic queries, and processes) can be used by the technical WPs of SOA4All to validate their concepts and prototypes.

8. References

[Becker2004] Becker, J. and Algermissen, L. and Niehaves, B. Organizational Engineering in Public Administrations –A Method for process-oriented eGovernment projects. Proceedings of the 2004 ACM symposium on Applied computing. NY, USA : ACM New York, 2004.

[Becker2009] Becker, Jörg, Algermissen, Lars and Falk, Thorsten. Prozessorientierte Verwaltungsmodernisierung. 2., überarb. u. erw. Aufl. s.l. : Springer, 2009. pp. XX, 344 S.

[Cardoso2004] Cardoso, J. Quality of service for workflows and Web service processes. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 1, pp. 281-308, 2004.

[Cardoso2008] Jorge Cardoso, Konrad Voigt, and Matthias Winkler. Service Engineering for the Internet of Services. Enterprise Information Systems, Lecture Notes in Business Information Processing (LNBIP). 2008.

[Denhardt2007] Denhardt, R.B. The new public service: Serving, not steering. s.l. : ME Sharpe Inc, 2007.

[DiNitto2009] Di Nitto, Elisabetta and Karastoyanova, Dimka and Metzger, Andreas and Parkin, Michael and Pistore, Marco and Pohl, Klaus and Silvestri, Fabrizio and den Heuvel, Willem Jan. S-Cube: Addressing Multidisciplinary Research Challenges for the Internet of Services. Towards the Future Internet: A European Research Perspective. Amsterdam : IOS Press, 2009.

[Ehm2009] Ehm, Peter, Performance Improvement in Public Sector Delivery. EPSA. 2009. Research Report.

[Frolick2006] Frolick, M.N. and Ariyachandra, T.R. Business performance management: One truth. Information Systems Management. 23, 2006, Vol. 1.

[Hermann2008] Hermann J. Schmelzer, Wolfgang Sesselmann. Geschäftsprozessmanagement in der Praxis. Münschen : Cark Henser Verlag, 2008. 978-3-446-41002-2.

[Hood1991] Hood, C. A public management for all seasons? Public administration. 1991, Vol. 69, 1.

[IoS2010] SAP Research. http://www.internet-of-services.com/. [Online] 2010. http://www.internet-of-services.com/index.php?id=55.

[Kaplan2001] Kaplan, Robert S., Strategic Performance Measurement and Management in Nonprofit Organizations. Jossey-Bass, 2001.

[Kopecky2008] Jacek Kopecky, Tomas Vitvar, Florian Fischer (2008). CMS WG Deliverable D11 v0.3 WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web. Working draft. June 2009, STI Innsbruck.

[Lambert2010] Dave Lambert and Carlos Pedrinaci (2010). CMS WG Deliverable WSMO-Lite Extras: The Minimal Service Model and Service Templates. Working draft. January 2010. STI Innsbruck

[Leitner2003] Leitner, Christine. eGovernment in Europe: The State of Affairs. s.l. : eGovernment in Europe: The State of Affairs, 2003. ISBN 13 978-90-6779-182-3.

[Lytras2006] Lytras, M.D. The Semantic Electronic Government: knowledge management for citizen relationship and new assessment scenarios. Electronic Government, an International Journal. 2006, Vol. 3.

[Moore2003] Moore, M.H. and COSKUN, A. The Public Value Scorecard: A Rejoinder and an Alternative to'Strategic Performance Measurement and Management in Non-Profit Organizations' by Robert Kaplan. The Hauser Center for Nonprofit Organizations. 2003.

[Niven2008] Niven, Paul R. Balanced scorecard step-by-step for government and nonprofit agencies / Paul R. Niven. 2nd ed. . s.l. : J. Wiley \& Sons, Hoboken, N.J. : , 2008 . pp. xvii, 365 p.

[Oaks2003] P. Oaks, A.H.M. Hofstede, D. Edmond. Capabilities: describing what services can do. Proceedings of Service-Oriented Computing – ICSOC 2003. s.l. : Springer, 2003.

[OSullivan2006] O'Sullivan, Justin. Towards a Precise Understanding of Service Properties. Queensland University of Technology. 2006.

[Paolucci2002] Paolucci, M., Kawamura, T., Payne, T. R., & Sycara, K. P. (2002). Semantic Matching of Web Services Capabilities. In Proceedings of the 1st International Semantic Web Conference (ISWC 2002), Sardinia, Italy, volume 2342 in Springer Lecture Notes in Computer Science (pp. 333–347)

[Roman2005] Roman, D., Keller, U., Lausen, H., de Bruijn, J., Stollberg, M., Bussler, C., Fensel, D., et al. (2005). Web Service Modeling Ontology. Applied Ontology 1, p77-106, 2005

[Sattler2005] Sattler, U., Motik, B. and Hustadt, U. 2005. Data Complexity of Reasoning in very Expressive Description Logics. In proceedings of 19th IJCAI pp.466-471, Morgan Kaufmann

[Sbodio2007] Marco Luca Shodio and Claude Moulin (2007). SPARQL as an expression language for OWL-S. In OWL-S: Experiences and Directions, a Workshop at the 4th European Semantic Web Conference (ESWC 2007). June 2007

[Snabe2009] Snabe, J. H., Rosenberg, A., Moeller, C., & Scavillo, M. (2009). Business Process Management the SAP Roadmap. ISBN 978-1-59229-231-8, SAP Press Series and Galileo Press, 1st ed.

[Stötzer2009] Stötzer, Sandra. Stakeholder Performance Reporting von Nonprofit Organisationen. s.l. : Gabler, 2009, Das System Nonprofit Organisation Umfelder,Stakeholder und Steuerung, pp. 9-220.

[Tao2004] Tao, Kwei-Jay Lin. s.l. : Service Selection Algorithms for Web Services with End-to-End QoS Constraints. IEEE Computer Society, 2004, E-Commerce Technology, IEEE International Conference on, Vol. 0, pp. 129-136.

[Vitvar2008] Tomas Vitvar, Jacek Kopecky, Dieter Fensel (2008). WSMO-Lite Annotations for Web Services. In proceedings of 5th European Semantic Web Conference (ESWC 2008). P.674-689. June 2008

[Vitvar2009] Tomas Vitvar, Jacek Kopecky, Dieter Fensel (2009). MicroWSMO and hRests: SOA4All D3.4.6. UIBK 2009

[Ziekow2009] Ziekow, Jan. Herausforderung e-Government. s.l. : Prof. Dr. Hermann Hill, Prof. Dr. Dieter Engels, 2009, Vom Verwaltungsverfahren über den Geschäftsprozess zum IT-Workflow, pp. 69-87.

Annex A: Related Work

Several EU and national projects have already developed ontologies that may be reused in the context of this WP. In addition to well-known ontologies such as Dublin Core and SIOC, some ontologies have been developed specifically for public administrations. In this Annex, we provide a small survey of existing work and an analysis of what may be reused.

KnowledgeWeb

Project	EU FP6, 01/2004 – 12/2007
URL	http://knowledgeweb.semanticweb.org/semanticportal/sewView/frames.html
Executive Summary	The Knowledge Web project supported the transformation process of semantic technologies from academia to the industry. It focused on three areas: With the help of industrial participation, semantic technologies should be disseminated more quickly in Europe and find practical applications. Through the integration of the education sector, the network EASE (European Association for Semantic Web Education) was established to promote educational activities in the semantic field. In addition, semantic web research activities have been coordinated by the project. The findings of the project cover five main areas in particular. It has been researching on ontology language extensions, which establishes inter alia an OWL extension and laid the foundations for the invention of SPARQL. The life cycle of an ontology has been further explored, which includes creation, versioning, evaluation, and negotiation.
Ontologies	Ontologies were reused from the projects OntoWeb and Esperanto portal. With industry participation, a portal ontology has been developed, which is available in RDF and OWL. Detailed description of the portal ontology: http://knowledgeweb.semanticweb.org/semanticportal/deliverables/D1.6.2.pdf
Formalisms	OWL, RDF
Analysis	KnowledgeWeb has not developed a Public Sector or a general service ontology. The portal ontology contains only a few standard concepts, which could be reused for a service ontology, but these concepts are available in even more detailed form other projects. For this reason, the existing ontologies of KnowledgeWeb are not used in this deliverable.

STASIS

Project	EU FP6, 08/2006 - 09/2009
URL	http://www.stasis-project.net/
Executive Summary	The project STASIS (Software for Ambient Semantic Interoperable Services) focused on the problems in electronic commerce of SMEs. Thereby especially the exchange of information between different

	companies and companies and the public administration is considered. The issue concerns the format of the information that cannot be understood by all trading partners. In practice, information usually is assigned based on its syntax. This mapping causes some disadvantages, such as error-proneness, lack of reusability of the software components, or the expensive purchase of IT-experts. STASIS offers business users as its target group a collection of electronic tools. The collection provides easy access for analysis, display, comparison and use of semantics to improve the business processes of various trading partners. The mapping is performed based on semantics. Semantic concepts are defined and linked with the help of ontologies. The user is enabled to identify the semantics of a schema and map it semi-automatically to the schema of the business partner. Created mappings are stored in a distributed directory which allows the reuse and integration of already generated mappings. The project has been tested in the furniture and automotive industries which are typical areas of a large variance in terms of infrastructure and data formats. As a main result of the project STASIS provides some services and plug-ins available as a desktop application based on the Eclipse platform.
Ontologies	OWL Encodings of five main ontologies: • EDI - Electronic Data Interchange • FF - Flatfile Structure • LDM - Logical Data Model • RDB - Relational Database • XML - XML Schema Description
Formalisms	OWL-DL
Analysis	The emphasis of the project STASIS is on the technical implementation of the ontology mapping based on the language OWL-DL. Thus, ontologies are used to map the information from the source system to the target system. For this purpose the respective business scheme is internally mapped into a neutral format. The use cases focus on the automotive and furniture industries. SOA4All uses WSML as an ontology language. With the help of WSMO Studio, OWL-DL based ontologies can in principle be transformed into WSML but requires significant post-editing overhead. Concrete models developed by STASIS cover mainly technical and structural information from SMEs, as well as specific content information of the automotive and furniture industries were transferred into ontologies, which cannot be used for WP7.

TEXO

Project	Part of German BMBF project THESEUS, 02/2007 - 01/2011
URL	http://www.theseus-programm.de/anwendungsszenarien/texo/default.aspx

Executive Summary	TEXO is an integrated part of the THESEUS research program which focuses on developing new technologies for the internet of services. TEXO deals with the change of society based on services from an industrial to a dynamic non-linear value chain. Through networked and flexible services there will arise new business models particularly in the areas of development, design, production, marketing, and promotion. In a marketplace supply and demand of Web-based services come together. The client receives greater transparency about the services and businesses can offer their services and reach entirely new customer groups. Also services from multiple vendors can be combined with each other. TEXO aims to explore and develop an infrastructure, which with the services can be found, combined, used and paid for it on the Internet. Technically TEXO is based on SOA architecture. Thus, individual applications can be combined and the user is provided a service. Main results from TEXO will be the TEXO Service Marketplace. This online platform allows the trading of existing and new services. Based on standards for the description of technical services TEXO integrates the development of the USDL language.
Ontologies	Service Ontology
Formalisms	USDL, OWL
Analysis	USDL is used to describe non-technical services in order to describe in addition to purely technical aspects, administrative and operational characteristics. The developed service ontology of TEXO is linked to the USDL language elements. A similar approach will be reused in the business semantic section. In addition, a continuous exchange takes place between the ongoing project TEXO and SOA4All. Furthermore the USDL research findings of a SAP transfer project can be integrated to this deliverable.

SUPER

Project	EU FP6, 04/2006 - 03/2009
URL	http://www.ip-super.org/
Executive Summary	The SUPER project aims to improve modeling and managing of business processes through the integration of semantics with BPM. To combine the aspects of semantic Web services and BPM, an architecture with the following three layers has been developed: Process Modeling layer, Process Engine layer and Transport Layer. SUPER aims to improve the cooperation between the business level and technical level, so that business process management functions can also be carried out by business experts.
Ontologies	Within the project a variety of ontologies have been developed (see WSML ontology download: Organization, Process, sRBE, Domain (confidential, but

<table>
<tr><td></td><td>for telecommunication sector):
1) Organizational Ontologies

bfo - Business Function Ontology
Business Activities, Delivery Management, Activity
bmo - Business Motivation Ontology
Strategy, Mission, Course, Action
bronto - Business Roles Ontology
Competitor, Manufacture, Buyer, Business Partner
deo - Policy Decision Event Ontology
Input Event, AND Operator, State, Complex Event
dro - Business Policy Rules Ontology
Process Rule, Duty Rule, Structure Rule, Rule Configuration
ecao - Event Condition Action Ontology
Filter, Action, Fallback, Custom Action
oso - Organizational Structure Ontology
Organization Unit, Business Function, Person
ouo - Organizational Unit Ontology
Permanent Organization Unit, Project Unit, Finance Department
resont - Process Resource Ontology
Primary Resource, Consumed Resource, Destroyed Resource, Shared Input

2) Process Ontologies

bpel20 - BPEL Ontology
bpmo - Business Process Management Ontology
bro - Behavioural Reasoning Ontology
sbpel - Semantic Extensions to BPEL
sbpmn - Business Process Modeling Notation
sepc - Event Driven Process Chain Notation
upo - Upper Level Process Ontology
WSDLExtension4BPEL

3) sRBE

BQR - Business Question Repository
RBEO - Reverse Business Engineering Ontology
</td></tr>
<tr><td>Formalisms</td><td>WSML</td></tr>
</table>

Analysis	SUPER provides a variety of WSML ontologies which are freely available to the majority on the website of the project. Most of the developed ontologies describe the organizational structure of enterprises. In addition, process and domain ontologies were produced, but which are not public sector specific. This kind of ontologies is not of great importance for WP7.6 and will not be reused.

SemanticGov

Project	EU FP6, 01/2006-04/2009
URL	http://www.semantic-gov.org/
Executive Summary	The project SemanticGov aims to build an infrastructure for the public administration, particularly software, models, and services, to enable the offering of semantic Web services in this domain. By creating an infrastructure, the interoperability between the various offices of public administrations can be improved within a country and at EU level. Shared ontologies and semantically annotated services support the semi-automatic process integration. For customers, it simplifies the handling of processes shared by several offices. In addition, existing processes should be improved or redesigned with this infrastructure. From a technical point of view, SemanticGov combines the paradigm of SOA with semantic Web services to model the domain of public administration. The developed infrastructure is divided into services and processes, models and ontologies, as well as applications and portals. The solution includes creation, discovery, composition, access, and monitoring of eGovernment services. The developed service model of the project SemanticGov has been used by the Ministry of Finance of Cyprus in order to document more than 100 different public services.
Ontologies	There have been developed several ontologies for the public administration. • GEA, Enterprise Architecture Governance: Generic domain model which is considered as a reference model for e-governance domain. • PA domain ontology: Supports specific use cases of the project • PA service ontology: reference ontology for the PA services • More use case specific ontologies
Formalisms	WSML, WSMO-PA (This is a developed specification to better allocate between the GEA and WSMO.)
Analysis	The project SemanticGov developed several ontologies specifically for the domain of public administration. There is a domain and a service ontology implemented in WSML that could be a basis for WP7 ontologies. Also the GEA ontology could be reused for the service classification. However, none of the developed ontologies are publicly available. An evaluation of the research information was just possible with the help of the project

	deliverables. But the GEA Service Model for public administration was adapted and reused from the project Access eGov. The developed ontologies of Access eGov could be reused in WP 7.6. [see section 0]

OntoGov

Project	EU FP6, 01/2004-06/2006
URL	http://km.aifb.uni-karlsruhe.de/projects/ontogov
Executive Summary	The project OntoGov has the goal to create a framework for the operation of eGovernment services. The main focus is on the process lifecycle and concerns on the one hand the consistency of a process creation and on the other hand the expressibility of a process. OntoGov thus addresses the problem of existing systems that often are either specialized in process modeling or process feasibility and combines these two aspects and also enables integration with external business applications. The developed framework provides an integrated modeling, configuration and orientation of services for the eGovernment domain. With the help of the developed ontologies a high degree of standardization and automation can be achieved. Through web services an open and flexible architecture is created which is independent of the platform or programming language and supports interoperability between internal and external services.
Ontologies	Domain-oriented ontologies • Legal • Organizational • Domain • Process • Live Event • Lifecycle • Profiles Administration ontologies • Service Evolution • WSOR Service
Formalisms	OWL-S (based upon OWL)
Analysis	OntoGov provides a set of ontologies for the modeling of services in the public administration. We assume that the domain, process, live event and lifecycle ontology of the domain orientated ontologies have great potential to be mapped to WSML. Currently, the OntoGov ontologies are not available and the content is not described in detail in the available source. An important point in evaluating the usability of the ontologies is that the

	domain-specific semantic annotations in the OWL-S model are generally limited. Moreover, it should be investigated whether the models focus exclusively on technical aspects. Parts of the Service and Web Service Orchestration ontology were reused within the Access e-Gov project (see below).

DIP

Project	EU FP6, 01/2004 - 12/2006
URL	http://dip.semanticweb.org/
Executive Summary	The project DIP aimed to extend the semantic Web and Web service technologies so that services and data can be searched and found in an environment and are automatically linked. It had four main objectives: • open source architecture for semantic Web services • use case implementations of SWS in different sectors: telecommunications , eGovernment, eBanking • practical, exploitable tools and methods that can be deployed for eWork, eGovernment, and eCommerce • impact on international standards through member submissions to the World Wide Web Consortium (W3C), the OASIS Semantic Execution Environment (SEE) Technical Committee, and OMG.
Ontologies	An e-Government ontology was developed from the so-called seamlessUK taxonomy.
Formalisms	WSML, WSMO
Analysis	The developed ontology is UK-specific and highly focused on the organizational structure. This DIP approach does not match with the SOA4All use case that is neither organization nor country-specific. For this reason, the DIP eGovernment ontology could not be reused for the ontology concepts of this deliverable.

Access-eGov

Project	EU FP6, 01/2006 - 03/2009
URL	http://www.accessegov.org/
Executive Summary	Access-eGov aims to improve the access for citizens and businesses to provide services of public administration. There for existing electronic and traditional services shall be better combined. Access-eGov wants to handle the existing problems with services. Information shall be made available through a single access point, so that it can be combined and integrated by

	a virtual assistant. Businesses and citizens will be provided two different types of services. Context dependent the user is provided a variety of eGovernment services and traditional services. In addition, a scenario is created, that regularly integrates both types of services. A virtual personal assistant accompanies the user through the scenario. Another focus of the project is the integration of disadvantaged user groups who receive a large benefit by the created application. The aim of the project is to provide services locally, regionally, nationally and across Europe to improve the interaction between eGovernment services. Based on semantic Web technology, Access-eGov has developed an open source solution that offers citizens and business users two types of services: 1. Meta-Service: depending on the needs and context of the user, eGovernment services are found 2. Scenario generation: after services have been found, a scenario is created and a personal assistant guides the user through the steps.
Ontologies	Three main types of e-government ontologies have been developed and validated: • AeG Core Ontology • Live Event Ontology • Domain Ontology
Formalisms	WSML, WSMO
Analysis	The developed domain ontology can be used as part of SOA4All. The ontology concepts are created very detailed and cover some standard scenarios of public administration. Access e-Gov reused the ontologies of related projects OntoGov and SemanticGov which were further developed. The domain-specific ontology has been newly developed and tested with three different use cases. As part of this deliverable large parts of the Access e-Gov domain ontology can be integrated in the WP7 specific SOA4All domain ontology.

Annex B: Optimized Service Selection for Processes

The optimized service selection problem is often discussed in the literature. Most works about on automated service selection focus on criteria such as quality of service (QoS) and consider only a small set of quality of service parameters like execution time, cost, availability and reputation for the optimization of the service selection. Using USDL-annotated services enables us to extend the set of non-functional service properties so that a large amount of process properties like the ones described in the process scorecard can be applied in the service selection phase. According to the composition functions analyzed in our KPI -model, there are three major classes of composed metrics for the process, the sum, the product and the quantity of the instances meeting a certain requirement. From the different approaches to model the problem of having a certain set of process activities *PA*, a set of available services *S*, bringing the needed functionalities and searching for the appropriate service combination, which best corresponds to the business requirements, we chose the graph approach. Applying graph algorithms to handle the problem is a solution, in which the algorithms need to be run only once to find the optimal service, whether they are structured as pipelines or directed acyclic graphs. The idea is to build all possible execution paths in the graph and to use a standard graph algorithm like e.g. a shortest path algorithm to select the best fitting services, in order to optimize a certain process indicator [Tao2004].

The graph construction can be done in the following manner:

1. Each service from the result set of the functional service discovery *S* is represented as a node in the graph; an initial start node and an initial end node are modeled as in the LPML (see D6.3.2)
2. The value of the analyzed service property is entered at the node, representing the service or at an outgoing edge; for the initial and end node the value 0 is entered.
3. From the initial node edges of the available services for the next activity of the process are modeled. Outgoing edges from all these nodes to the services available for the next activity are created and values are entered. For XOR-gateways edges have to be built to a set of all services in *S* for the process activities, following the gateway, whereas an and-gateway can be modeled as a sequence of the activities, following the gateway [Cardoso2004].

The following example demonstrates the graph approach: Consider the process depicted in Figure 20 is modeled with the SOA4All Composer (see D2.6.1).

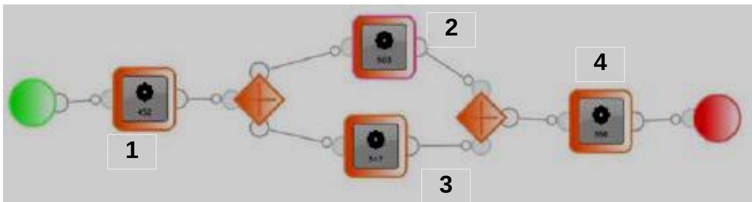

Figure 20: Sample Process

The set of process activities *PA* is given by *PA*={1,2,3,4}. A functional service discovery is executed and the following set of available SOA4All services is found:

- 1: S1 and S2
- 2: S3 and S4
- 3: S5, S6, and S7

- 4: S8

So that the set of available services *S* is *S*={S1,S2,S3,S4,S5,S6,S7,S8}. Assuming that the gateway in the workflow after activity 1 is a XOR-gateway and that the service property we are interested in is the service cost, given by c_i: $i \in \{1,..,8\}$, the graph representation looks like as given in Figure 22, whereas an AND-gateway (as is the case in Figure 20) would result in the graph depicted in Figure 21. Please note that for a parallel-gateway we do not consider any kind of sequence, because it does not affect the optimization algorithms.

Finding the optimal execution path (the service combination for the process activities) can be solved by using a shortest path algorithm for a directed graph like e.g. the Floyd-Warshall algorithm or the Bellman-Ford algorithm. The Floyd-Warshall algorithm computes the shortest path between all pairs of vertices. This algorithm can find an optimal solution with n^3 comparisons where n is the number of nodes, which is quite remarkable, considering the possibility that there may exist n^2 edges in the directed graph. For service properties, whose composition function is the product of all single values, the algorithm can be adapted in an appropriate manner. However, for the computation of the number of different instances, fulfilling a certain condition, an exhaustive search of all possible execution pads is needed.

In the envisioned demonstrator, we will provide the business user with the ability to select an enhanced service combination for a certain key performance indicator and to monitor the modification of a set of performance indicators when another service composition, so that he/she is able to select the services according to his/her preferences value for the performance indicators. In addition to this an optimization possibility for two indicators is proposed. We can model this problem as multiple-choice knapsack problem. Having a set of m business activities $\{a_1,\ldots,a_m\}$ such that each activity contains n_i services from the services set S. We use the variable x_{ij} to indicate whether the j-th service of the list of available services for the activity a_i is selected or not. The selection problem can be then modeled in the following way:

$$\min C = \sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij}$$

$$subject\ to \sum_{i=1}^{m} \sum_{j=1}^{n_i} t_{ij} x_{ij} \leq T$$

$$\sum_{j=1}^{n_i} x_{ij} = 1\,, i = 1, \ldots m;\ x_{ij} \in \{0,1\} \bigvee i,j$$

The values c_{ij} are the values of the first performance indicators selected by the business user, e.g. the price, whereas the values t_{ij} are the ones of the second parameter, e.g. the execution time. The value for the maximal acceptable execution time need to be inquired from the user and the aggregation function can as well be replaced by max.

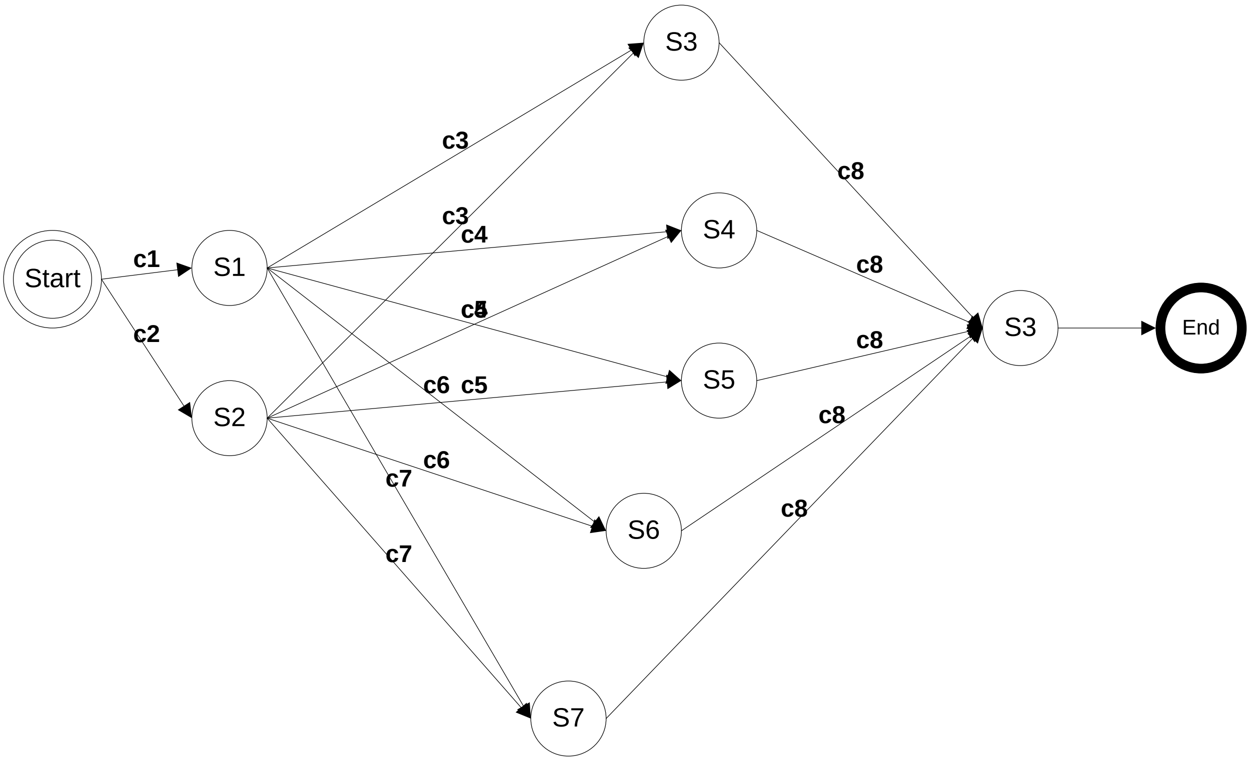

Figure 22: Graph for Service Selection Problem (XOR)

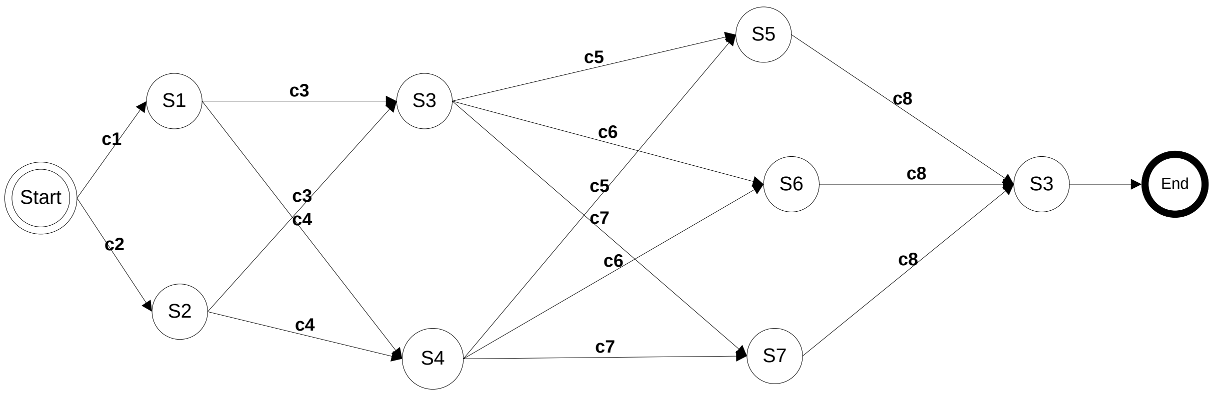

Figure 21: Graph for Service Selection Problem (AND)