Knowledge modelling for integrating semantic web services in e-government applications

Conference or Workshop Item

Knowledge Modelling for Integrating E-Government Applications and Semantic Web Services*

Alessio Gugliotta, Liliana Cabral and **John Domingue**
Knowledge Media Institute, The Open University
Walton Hall
Milton Keynes, MK7 6AA, UK.
{a.gugliotta, l.s.cabral, j.b.domingue}@open.ac.uk

Abstract

Service integration and domain interoperability are the basic requirements in the development of current service-oriented e-Government applications. Semantic Web and, in particular, Semantic Web Service (SWS) technology aim to address these issues. However, the integration between e-Government applications and SWS is not an easy task. We argue that a more complex semantic layer needs to be modeled. The aim of our work is to provide an ontological framework that maps such a semantic layer. In this paper, we describe our approach for creating a project-independent and reusable model, and provide a case study that demonstrates its applicability.

Introduction

The current trends in e-Government applications call for joined-up services that are simple to use, shaped around and responding to the needs of the citizen, and not merely arranged for the provider's convenience. In this way, the users need have no knowledge of – nor direct interaction with – the government entities involved. On practical grounds, the *integration* of services is a basic requirement of *service-oriented systems*, which aim at gathering and transforming processes - needed for a particular user - into one single service and the corresponding back-office practices. They enable the building of agile networks of collaborating business applications distributed within and across organization boundaries. Thus, services need to be *interoperable* in order to allow for data and information to be exchanged and processed seamlessly across government.

The Semantic Web (T. Berners-Lee 2001) aims to alleviate integration and interoperability problems. By allowing software agents to communicate and understand the information published, the Semantic Web enables new ways of consuming services. In particular, Semantic Web Service (SWS) technology provides an infrastructure in which new services can be added, discovered and composed continually, and the Public Administration (PA) processes automatically updated to reflect new forms of cooperation (Gugliotta *et al.* 2005). It combines the flexibility, reusability, and universal access that typically characterize a Web Service, with the power of semantic mark-up, and reasoning in order to make feasible the invocation, composition, mediation, and automatic execution of complex services with multiple conditional paths of execution, and nested services inside them (Sycara *et al.* 2003), (Domingue *et al.* 2004).

However, the integration between e-Government applications and SWS's is not an easy task. We present an approach for knowledge management based on SWS technology and the following e-Government requirements:

- the PA worker - and in general a domain expert - does not directly use the SWS infrastructure to represent knowledge internally. For instance, organizations will likely adopt their own workflow paradigm to describe their processes (van der Aalst, ter Hofstede, & Weske 2003).
- The PA work routines involve interactions with non-software agents, such as citizens, employees, managers and politicians. Multiple viewpoints need to be considered.
- In real cases, component services are not atomic, and cannot in general be executed in a single-response step; they may need to follow an interaction protocol with non-software agents that involves multiple sequential, conditional and iterative steps. For instance, a service may require a negotiation between the user and the provider.
- Web service description is an important but restricted aspect of an e-Government service-supply scenario.

In this paper, we argue that a more complex semantic layer for managing government services needs to be modelled - and a middleware system designed on such a model - in order to meet the requirements of real-life applications. In particular, we identify three knowledge levels.

Configuration, describing the context in which services are supplied: requirements, resources, actor's role, business processes, and transactions of an e-Government service-supply scenario.

Re-configuration, describing the context in which services may be modified: legislations, policies, and strate-

*This work is supported by the Data, Information and Process Integration with Semantic Web Services (DIP). DIP is an Integrated Project (no. FP6 - 507483) supported by the EU's IST programme. – http://dip.semanticweb.org.

gies influencing the development and management of an e-Government service-supply scenario.

Service delivery, adopting SWS technology as the base for the description, discovery, composition, mediation, and execution of (Web) services.

As a result, the integration of e-Government applications with SWS's requires a framework which maps and combines the knowledge levels described above. The aim of our work is to provide such a framework with which most PA's – or generally organizations – can identify, from which they can work when designing and delivering e-Government services. Such a general framework can be adapted and applied as appropriate.

Our approach is grounded on a technological paradigm able to fit a general distributed organization of knowledge, with focus on the supply of services. The proposed framework is considered from the following two different dimensions.

(i) *Conceptual modelling*: this is a double stage process that first creates a conceptualization of the reality in terms of conceptual models, and then uses ontologies to represent the semantic structure of involved knowledge, enabling knowledge use and reuse. The result is an *ontological framework for service-oriented e-Government applications*.

(ii) *Creating of an infrastructure for semantic interoperability*: software modules are used to implement the functionalities of a middleware system that enables the automated interpretation and paves a common ground for services. The result is a *semantically-enhanced middleware for service-oriented e-Government applications*.

Current work concerns the first dimension, on which we shall focus in the rest of the paper.

Related Work

Although service-oriented computing is a relatively new field, many e-Government applications have been developed and various approaches have been proposed.

To quote a few examples, *eGov* (eGOV 2004), and *EU-PUBLI.com* (EU-PUBLI 2004) define architectures based on Web services interfacing PA legacy systems. The goal of these projects is to achieve one-stop E-Government. XML dialects are used to define metadata and orchestration of services.

SmartGov (SmartGov 2004), *ICTE-PAN* (ICTE-PAN 2004) and *E-Power* (Engers *et al.* 2004) projects use ontologies for representing e-Government knowledge. In particular, SmartGov and ICTE-PAN developed two ontologies describing the profile of a service.

OntoGov (OntoGov 2004) and *TerreGov* (TerreGov 2004) adopt SWS approach for describing services provided by PA's. However, they do not completely take advantage of SWS technology. OntoGov develops an own ontology for describing e-Government services mixing aspects of the two main approaches OWL-S (OWL-S Coalition 2004) and WSMO (Dumitru, Holger, & Uwe 2004), in order to satisfy some e-Government requirements that are not addressed by such approaches. TerreGov adopts OWL-S for describing, and discovering services but it uses BPEL language for describing composition of services (eProcedure).

These approaches face more or less the same problems: there is no generic domain analysis for the overall PA system at any level of granularity; there are no generic PA models for processes and objects; there are no ontologies for modelling PA objects and relationships; there are no standard vocabularies for describing concepts. Consequently, the researchers have to build from scratch PA ontologies to be used as test-beds for demonstrating the functionalities of their systems. The main focus of these initiatives is not to build a PA domain ontology, but rather to test and validate specific technological solutions. As a result, they propose ad hoc description for the PA domain and far from being considered reusable.

Moreover, existing approaches usually address specific service-oriented models, where the provider's point of view plays a central role. However, the e-Government scenario is composed by several actors. Each of them deals with different kinds of knowledge, conceptions, processes; in other words they have different viewpoints. Such viewpoints influence and relate to the service differently.

Requirements for the Conceptual Model

Starting from the analysis of the above projects, we defined the following objectives of our approach.

General purpose. The aim of our work is not to represent all of the existing concepts and relations connected to the e-Government domain. As in the ICTE-PAN project, the idea is to create some modules driving domain experts to develop domain ontologies describing the specific scenario and helping developers to implements SWS's based on the domain expert's representations. In particular, these modules outline a generic service-supply scenario that domain experts can adapt and extend using different levels of granularity on the basis of scenario characteristics. The result is a re-usable, extensible, and flexible model.

Life Event approach. All of the projects introduced here adopt a service-oriented approach. The service provided by organizations is the central concept. For instance, in the eGov and OntoGov projects, the user point of view is described by a taxonomy of life events simply representing how to arrange the services in the portal. In our vision, the life event concept plays a central role prompting the supply of services by several organizations and representing the point of contact among the different actor's viewpoint. It represents the starting point for the description of the involved scenario knowledge.

Contextualization. Our approach allows us to contextualize – i.e. describing various notions of context, non physical situations, topics, plans, beliefs, etc. as entities – an e-government scenario in terms of descriptions. In particular, we distinguish between descriptive entities – that are independent views on a scenario by different involved actors – and the actual objects they act upon - representing the concepts of the actor's vocabularies. This captures that multiple overlapping (or alternative) contexts may match the same world or model, and that such contexts can have systematic relations among their elements.

PA Autonomy and Cooperative Development. The domain standardization (introduced by the different e-Government projects) can help, but it does not necessarily unify the aims and languages of all the involved organizations and actors. Each of them should keep its autonomy describing its own domain. Actually, distinct organizations could use or describe the same concepts differently. This implies the need of address the issues of mediation between heterogeneous sources, but allows the co-operative development of an e-Government application.

Business Process and Interaction description. The process flow of e-Government processes can be modeled using standard control structure and tasks. Different projects adopt different approaches. For instance, OntoGov adopts the OWL-S process model while TerreGov will adopt BPEL. Unlike the other approaches, we introduce an Interaction description that is a useful means of introducing model checking to the requirements gathering process, as well as a key but too often neglected component of business process. Actually, we distinguish between a *plan* describing processes and organising concepts within an actor's viewpoint and *interaction* describing mutual actions involving two different actor's viewpoints.

Delegation. Service integration will allow organizations to delegate the execution of some of their tasks to external organizations. This includes looking for and identifying the right organization. In our approach, we explicitly define how declare delegate tasks. This aspect is strictly connected to the above interaction description representing the protocol to consume the delegation.

SWS standards. SWS technology addresses the integration and interoperability issues between services provided by heterogeneous organizations. However, some e-Government requirements cannot be represented by existing SWS approaches. In our approach, we clearly distinguish between the e-Government scenario description – addressing the e-Government requirements – and SWS descriptions. The two levels are integrated without requiring changes to SWS standards.

Meta-Modelling the Conceptual Model

A conceptual model is an abstract and simplified description of the reality that has to be represented. An explicit specification of a conceptual model is an ontology.

To match the re-usability requirement of our approach, we refer to the conceptual model as an abstract definition of how to describe and develop a domain of interest: a model of modelling (Fernandez-Lopez 2001). It points out the building blocks that are used in models of the domain, the relationships between the building blocks, and how to build models.

The ontologies mapping such a conceptual model are domain-independent ontologies – i.e. meta-ontologies – specifying the schema to be followed by the modeling process and the general concepts and relations that may be extended and adapted.

Applying a specific scenario to the meta-model, the result is a model for a specific application (Figure 1). Starting from meta-ontologies, the resulted ontologies describe application-dependent concepts, relations, axioms, etc.

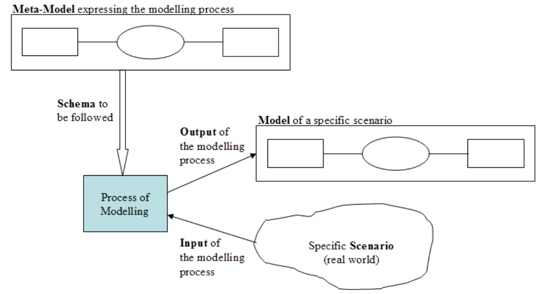

Figure 1: Meta-modelling approach.

The meta-modelling approach is the base for the *cooperative and distributed* development of an application-specific conceptual model. It allows involved actors to keep their autonomy in the description of their domains: each actor follows the proposed schema (meta-graph) to create ontologies extending and adapting the meta-ontologies. All of the obtained ontologies describe the application-specific conceptual model; each of them can define one or more actor's viewpoints and may refer to other application ontologies

Ontologies for Meta-Modelling

To facilitate the building of meta-ontologies, we refer to existing reference ontologies. We extend them and reuse some of their modules to create our ontologies. Actually, we refer to DOLCE (Oltramari *et al.* 2002) as upper ontology for describing domain concepts, its *Description & Situation* module (Gangemi & Mika 2001) as approach for knowledge contextualization (i.e. representing various points of view on a scenario, possibly with different granularity), and WSMO (Dumitru, Holger, & Uwe 2004) for describing Web services.

Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)

DOLCE belongs to the WonderWeb project Foundational Ontology Library (WFOL) and is designed to be minimal in that it includes only the most reusable and widely applicable upper-level categories, rigorous in terms of axiomatization and extensively researched and documented (Oltramari *et al.* 2002).

DOLCE has been chosen due to its internal structure – rich axiomatization, modularization, explicit construction principles, careful reference to interdisciplinary literature, commonsense-orientedness. In addition, being part of the WFOL, DOLCE will be mapped onto other foundational ontologies – possibly more suitable for certain applications – and be extended with modules covering different domains (e.g., legal and biomedical); with problems and lexical resources (e.g., WordNet-like lexical). Internal consistency and external openness make DOLCE specially suited to our needs.

The Description & Situations (D&S)

D&S is the module of the DOLCE ontology that describes context elements. While modelling physical objects or events in DOLCE is quite straightforward, intuition comes to odds when we want to model non-physical objects such as social institutions, plans, organizations, regulations, roles or parameters. The representation of context is a common problem in many realistic domains from technology and society which are full of non physical objects, e.g. non-physical situations, norms, plans, beliefs, or social roles are usually represented as a set of statements and not as concepts.

D&S results to be a theory of ontological contexts because it is capable of describing various notions of context or frame of reference (non physical situations, topics, plans, beliefs, etc.) as entities. It features a philosophically concise axiomatization.

D&S introduces a new category, *Situation*, that reifies contexts, episodes, configurations, state of affairs, cases, etc. and is composed by entities of the ground ontology (e.g. a domain ontology derived from DOLCE). A Situation satisfies a *Situation Description*, which represents a conceptualization (as a mental object or state), hence generically dependent on some agent, and which is also social, i.e. communicable.

Situation descriptions are composed of descriptive entities, i.e., *Parameters*, *Functional Roles* and *Courses of Events*. Axioms enforce that each descriptive component links to a certain category of DOLCE (the actual objects they act upon): Parameters are *valued by* Regions, Functional Roles are *played-by* Endurants and Courses of Events *sequence* Perdurants.

This captures that multiple overlapping (or alternative) contexts may match the same world or model, and that such contexts can have systematic relations among their elements.

D&S shows its practical value when applied as reference ontology for structuring application ontologies that require contextualization. As we will see in the remainder of this paper, this is the case when describing the e-Government service-supply scenario.

Web Service Modelling Ontology (WSMO)

We adopt WSMO (Dumitru, Holger, & Uwe 2004) because it follows design principles that embrace our approach and that other standards do not have.

Strict Decoupling. Decoupling denotes that WSMO resources are defined in isolation, meaning that each resource is specified independently without regard to possible usage or interactions with other resources. This complies with our distributed and cooperative approach.

Centrality of Mediation. As a complementary design principle to strict decoupling, mediation addresses the handling of heterogeneities that naturally arise in distributed environments. Heterogeneity can occur in terms of data or process. WSMO recognizes the importance of mediation for the successful deployment of Web services by making mediation a first class component of the framework.

Ontological Role Separation. User requests are formulated independently of (in a different context than) the available Web services. The underlying epistemology of WSMO differentiates between the desires of users or clients and available Web services. This complies with our multi-viewpoint approach around the concept of life event.

Service versus Web service. A Web service is a computational entity which is able (by invocation) to achieve a goal. A service in contrast is the actual value provided by this invocation. Thus, WSMO does not specify services, but Web services which are actually means to buy and search services. This complies with our clear separation between the description of Web services and the context of e-Government service-supply where Web services are used.

The main components of WSMO are Ontologies, Goals, Web Services and Mediators.

Goals represent the objectives that users would like to achieve via a Web Service (WS). The WSMO definition of goal describes the state of the desired information space and the desired state of the world after the execution of a given WS. A goal can import existing concepts and relations defined elsewhere, by either extending or simply re-using them as appropriate.

Web Service descriptions describe the functional behavior of an actual WS. The description also outlines how Web Services communicate (*choreography*) and how they are composed (*orchestration*).

Mediators define mappings between components: for instance, a goal can be related to one or more web services through mediators. They facilitate the clear-cut separation of different interoperability mechanisms.

Ontologies provide the basic glue for semantic interoperability and are used by the three other components.

Mapping the Conceptual Model: the Ontological Framework

As the e-Government field involves several aspects, we do not refer to a unique conceptual model. Our work is founded on existing – and in some cases well known – conceptual models that define the elements of a government service-supply scenario, specify the actors and roles of an e-Government application, introduce the life event metaphor as the base for a multi-viewpoint approach, and describe the aspects of the e-Government business processes. The main elements described by these conceptual models are combined and mapped into a sound *ontological framework*. The latter introduces the following two clean separations.

(i) *Context vs SWS*. We distinguish between the description of the environment where the services are provided, used, and managed by different actors and the SWS's allowing the automatic discovery, composition, mediation, and execution of services. The former maps the e-Government application entities and requirements, and, particularly, the aspects that cannot be captured by SWS's such as interaction with non software agents, multiple viewpoints, distinct infrastructure to represent knowledge internally, negotiation between user and provider. It represents input for the latter that completes the scenario with the technical description of computable entities that are able to achieve a goal. This separation allows the integration of SWS standards and e-Government applications without affecting SWS standards.

(ii) *Context vs Vocabulary*. We distinguish between descriptive entities (independent views on a scenario by different involved actors) and the actual objects they act upon (representing the vocabulary of different involved actors). This separation allows to adopt distinct – and in some cases already existing – vocabularies for multiple viewpoints.

Figure 2 depicts the architecture of our framework and its three meta-ontologies.

Core Life Event Ontology (CLEO) is the heart of our framework allowing the description of the configuration and reconfiguration knowledge by multiple e-Government actors. It represents the *contextualization* of the scenario.

Service Ontology allows the description of the service delivery knowledge. Based on the above scenario description, developers may provide *SWS descriptions* addressing integration and interoperability issues; i.e. it completes CLEO by describing Web services and their composition and mediation.

Domain Ontology defines the *vocabularies* used by multiple actors in the description of their viewpoints. It represent the *lexical layer* of the framework.

Finally, other ontologies can be imported in order to extend or specialize the conceptual model. For instance, an existing ontology can be used to describe the specific vocabulary of an involved actor.

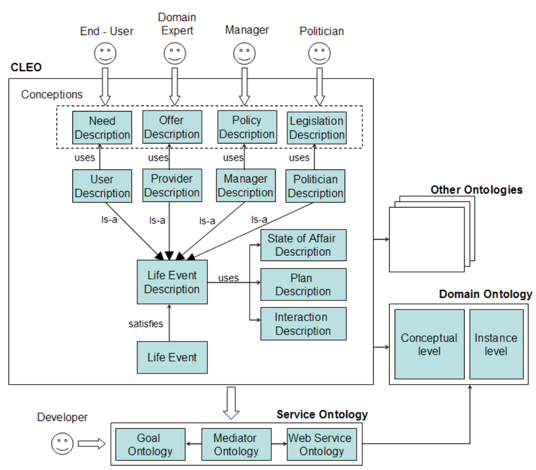

Figure 2: The proposed ontological framework

Core Life Event Ontology (CLEO)

As a main result of our work, we created a core ontology named Core Life Event Ontology (CLEO). A core ontology is a very basic and minimal ontology consisting only of the minimal concepts required to understand the other concepts. It should contain a number of generic concepts and method-independent definitions and it is extremely useful for reuse purposes.

The objective of our ontology is to enable e-Government actors to represent the knowledge they want to describe with the scenario in their own "language" and based on concepts which are familiar to them, excluding the use of technical concepts. In particular it allows to define what kinds of concept each participant-actor should contribute (e.g. need, offer, policy, and legislation), and which are the allowed associations-relations between these elements (e.g. influences, interactions, and plans).

It is a means for representing knowledge by multiple viewpoints and defining the context in which an application should be developed. The knowledge may include generic service processes as well as situation descriptions, involving human and not-human agents.

CLEO has been designed to be modular. Figure 2 shows the main modules from which it is composed. Every module can be readily extended and freely reused, and deals with one particular aspect of our conceptual model.

Life Event Description Module: it is the heart of CLEO, representing the life events and all actors' viewpoints that provide a description of them. It adopts the other modules to provide a sound description of the context around the life event.

State of Affair Description Module: gathers the elements (roles, attributes, and parameters) that are relevant in the description of a life event situation.

Conception Description Module: represents the conceptions that actors may describe in their viewpoint related to a specific state of affairs and life event: needs, offers, policies, and legislation.

Plan Description Module: allows the description of the tasks associated with the descriptions of a scenario and the organization into a plan within an actor's viewpoint.

Interaction Description Module: plays a double role, representing the interactions between two distinct viewpoints, and capturing possible mismatches in the description of exchanged resources. It is the unique module shared between two distinct viewpoints.

Finally, CLEO introduces a knowledge elicitation methodology that first helps domain experts to create a full description of a specific e-Government context using models close to their experience and specific languages of the different involved domains, and then drives the application developers to implement SWS descriptions eventually inferring knowledge from the context description.

The Domain Ontology

The Domain Ontology encodes concepts of the PA domain: organizational, legal, economic, business, information technology and end-user. They are the building blocks for the definition of CLEO and Service Ontology concepts.

Our aim is not to cover all the aspects connected with e-Government. Distinct PA's could use the same concepts differently or vice-versa adopt distinct terms for the same concept; a single PA may not share the same point of view and have different interoperability needs from other PA's. Multiple actors can use different vocabularies, also within the same organization. Standardization can help, but it does not necessarily unify the aims and languages of all the involved actors. It is important that every PA (or actor) keeps its autonomy in the description of its own domain (or viewpoint);

this does not affect our ultimate goals of interoperability and integration.

We designed a structure that resides on two level of abstraction: *conceptual* and *instance* level. Figure 3 shows four distinct ontologies (A, B, C, and D) derived from the conceptual level, ending and adapting the existing concepts and relations. They compose the instance level and are independent each other. Each of them describes a particular domain connected to a viewpoint; e.g. legislative terminology, technical terminology of an organization or a field, actor's language, manager's tasks, etc.

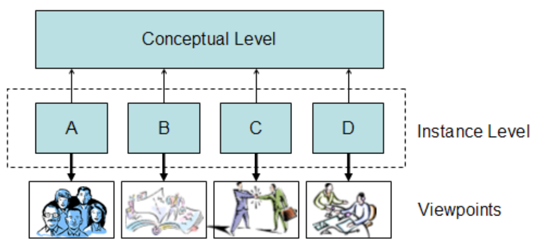

Figure 3: The two-levels structure of the Domain Ontology.

It is important to note that the aim of such a structure is to represent an heterogeneous scenario, and not create mappings between different concepts or solve existing mismatch problems. In our approach these tasks are delegated to the service delivery knowledge level; i.e. at service execution time, when the need for solving a mismatch arises.

The Service Ontology

The Service Ontology makes whole the representation of the scenario, modelling the *service delivery knowledge level* by means of the SWS technology. It allows the completion of the descriptions of (i) the services implemented by means of Web services, (ii) the e-Government processes that can be modeled as a composition of Web services (without user interaction), and (iii) the user requests of services. Moreover, it enables the specification of the mechanisms to solve existing mismatch problems at data and process level between distinct viewpoints. In other words, it represents the knowledge useful at runtime.

Because representing SWS's requires the use of technical concepts, the developers are responsible for the creation of this ontology. Concepts such as precondition, postcondition, grounding, orchestration and choreography of Web services are defined at this level. As we adopt the WSMO approach for SWS's, and this ontology is composed by three main modules: Goal Ontology, Web Service Ontology, and Mediation Ontology. Note that our work does not consist of improving existing SWS solutions, but enabling their application in the designed ontological framework.

Figure 4 depicts the intersection between the representations of the SWS's and e-Government applications context provided by the Service Ontology and CLEO, respectively. The two ontologies model two distinct knowledge levels that are provided by distinct kinds of actor, and obviously described by distinct modules. However, they intersect on two con-

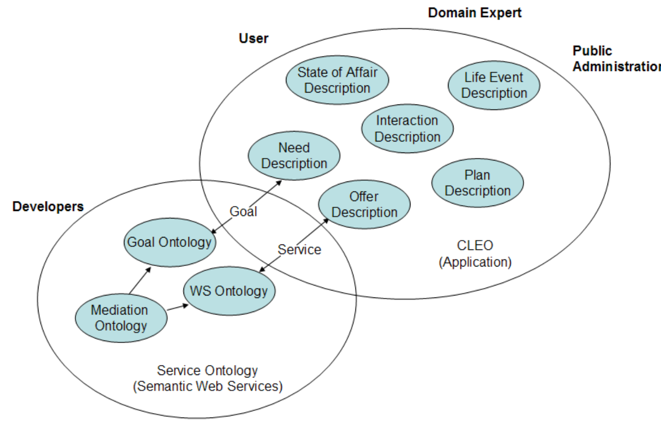

Figure 4: Intersection between the knowledge levels modeled by the Service Ontology and CLEO; the former represents the *service delivery* knowledge level (SWS), the latter the *configuration* knowledge level (e-Government application).

cepts: the goal and service descriptions. Such concepts are their points of contact, and, thereby, the way to actualize the integration between SWS and the e-Government application context descriptions. This conceptual overlapping allows the following two integration directions.

Using SWS descriptions within the context description: WSMO compliant descriptions of goal and services can be directly adopted within the context description for describing user's requests and Web services.

Inferring knowledge from the context for creating SWS descriptions: the description of the context where the service are supplied may represent the base for the definition of WSMO compliant goal and service descriptions. This also involves the definition of the mediation mechanisms.

To carry out the above integration purposes, we firstly derived the concepts of the Service Ontology from WSMO meta-models and then aligned such concepts to the CLEO meta-models (Figure 5). Axioms and rules enrich the Service Ontology for specifying the above alignment, and describe inference reasonings from CLEO descriptions in order to complete the WSMO compliant descriptions and handle mismatches. The Service Ontology may integrate further SWS approaches, simply adopting the same alignment mechanism.

A Change of Circumstances Case Study

We illustrate the elicitation methodology and the main elements of CLEO modules through an e-Government case study within the *change of circumstance scenario*. The prototype is a portal for Essex County Council in the UK (Drumm *et al.* 2005), (Cabral *et al.* 2005), where the following two governmental agencies were involved.

Community Care (Social Services) in Essex County Council: they typically have a coordinating role in relation to a range of services from a number of providers and special responsibility for key services such as support for elderly

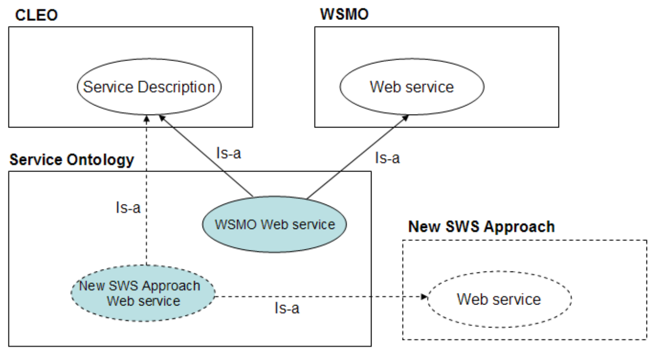

Figure 5: Aligning web service concepts of the Service Ontology: the class *WSMO-Web-service* is defined as sub-class of both CLEO and WSMO web service descriptions. Further SWS approaches simply adopt the same alignment mechanism.

and disabled people (day centers, transportation). It uses the SWIFT database as its main records management tool. *The Housing Department of Chelmsford District Council*: handles housing services and uses the ELMS database.

In this scenario, a case worker of the Community Care department helps a citizen to report his/her change of circumstance (e.g. address) to different agencies involved in the process. In this way, the citizen only has to inform the council once about his/her change, and the government agency automatically notifies all the agencies involved. An example might be when a disabled mother moves into her daughter's home. The case worker opens a case for a citizen who is eligible to receive services and benefits - health, housing, etc. Multiple service providing agencies need to be informed and interact.

The methodology

The proposed methodology improves the capture of e-Government service-supply scenario requirements and knowledge, in a robust and repeatable manner, whilst also eliciting an awareness of significant facets of the scenario much earlier during the knowledge capture phase.

The modules of CLEO represent the stages of the methodology that have to be followed and define the structure of the knowledge that have to be represented. CLEO indicates how real-life interaction scenarios can be decomposed and translated into models. The Domain Ontology will collect the terms extracted during the elicitation process, and the Service Ontology will contain the final result of the process: the SWS descriptions.

The methodology is summarized in the following stages:

1. *Life event and actor analysis*. The e-Government scenario is segmented along two orthogonal dimensions: life events, and actor's viewpoints. Segmentation allows to focus on a reduced and well-delimited sector of the scenario.

2. *Viewpoint analysis*. It represents the distribute and cooperative phase. Each identified actor independently defines its viewpoint on the life event. The adequate class *life-event-description* (user, provider, manager, or politician) has to be adopted for specifying the structure of the viewpoint. The following sub-stages are involved in this phase:

   - (a) *State of Affair analysis*. Main concepts of the domain are modelled as descriptive entities and used to describe the overall scenario of the problem that is being investigated.
   - (b) *Interaction analysis*. All of the interactions between couples of user-provider and provider-provider viewpoints are identified and described by means of the Interaction Description module.
   - (c) *Conception analysis*. The description of the scenario is improved by adding the conceptions (need, offer, policy, legislation) of the actors onto the state of affairs previously defined. In the cases of user and provider viewpoints, the defined conceptions refer to the defined interaction descriptions.
   - (d) *Plan analysis*. Describes the processes and dynamics within the viewpoint. Concepts connected to events and tasks are elicited.

   All of the identified concepts populate the *conceptual level* of the Domain Ontology, creating the specific *instance level* ontologies of the viewpoints.

3. *Model Specific Scenario*. Instances of the descriptions and concepts are created. In this way, the model is tested forcing a set of check axioms and rules to refine the representation.

4. *Create the SWS descriptions*. The obtained model is used as input of the SWS descriptions provided by developers.

Live event and actor analysis

The first stage is to examine the use case in order to identify the life events within it.

Our approach is based on the life event metaphor, which prompts the supply of services by PA's. We may simply consider how many different views exist on a life event: the citizen, the PA, the manager, the politician, etc. *Life-Event* and *Life-Event-Description* are center concepts of CLEO, respectively referring to the *Situation* and *Description* concepts of the D&S ontology.

Currently, we consider four kinds of life event description (user, provider, manager, politician), but further views may be added in the future when extension needs arise.

In the following we list the two life events considered in the case study.

Patient Moves House: A patient of the Social Services notifies that he/she changed address. This event triggers some changing of the information stored in the SWIFT and ELMS databases, and checking the eligibility of the patient to old and new services and benefits provided by the involved organizations. In case of eligibility of new services, a new patient assessment is necessary.

Patient Passes Away: A patient of the Social Services dies. The date of death should be set in the SWIFT database, and services and benefits have to be canceled.

In the rest of the dissertation, we refer to the first life event. The second stage of this phase is to define the actors that describe their viewpoints. In this case study two public administrations were involved.

Community Care (Social Services) in Essex County Council: typically has a coordinating role in relation to a range of services from a number of providers and special responsibility for key services such as support for elderly and disabled people (day centers, transportation). It uses the SWIFT database as its main records management tool.

The Housing Department of Chelmsford District Council handles housing services and uses the ELMS database.

Moreover, the end-user is the *case worker* of the Community Care department that helps citizens to report his/her changes of circumstance (e.g. address) to different agencies involved in the process. In this way, the citizen only has to inform the council once about his/her change, and the government agency automatically notifies all the agencies involved.

Viewpoint analysis

At this stage, we devise three teams for creating three descriptions according to their viewpoints: one user description for representing user requests, and two provider descriptions for representing available services. The three teams work independently building and using the respective lexical layers, and interfacing only to reach an agreement at the *Interaction analysis* stage. In this way, we can simulate the situation of distributed organizations that, driven by the framework, can autonomously describe their own domain.

Figure 6 shows that, for each domain, we refer to two ontologies: one that will contain the terms associated with the legacy systems (SWIFT, ELMS), and one that will contain other specific terms of the domain. All of the above ontologies will form the instance level of the Domain Ontology (Section). Without loosing generality, we assume that the case worker and community care viewpoints share the same ontologies.

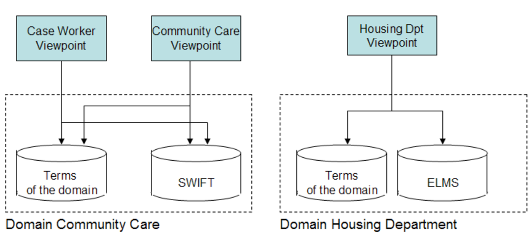

Figure 6: Reference domain ontologies for the considered viewpoints.

State of Affair analysis

The first step of the Viewpoint analysis is to identify the main concepts of the domain, and describe the states of affair where the services are requested and provided within the life event. More than one state of affairs can be identified within a life event: e.g. the initial and the final state of affairs. Each state of affairs defines the involved actors, resources, information, attributes, functional and non-functional parameters, and the relations among them. The concepts identified in this analysis enrich the defined Domain Ontology (Figure 6). On the basis of the descriptive entities used, we distinguish some sub-classes of the *State-of-Affair-Description*: e.g. *Service-Request*, defining a situation where an applicant requires services; *Processed*, defining a situation where one or more activities have been executed.

In the following, as an example, we summaries the analysis of the *case worker* viewpoint.

A *case worker* is involved in two main situations connected to the patient move house life event: (i) collecting patient information, and notifying his/her change of address; (ii) checking the patient eligibility to old a new services and benefits, and eventually opening a new patient assessment. These two situations have been mapped onto two couples of initial and final state of affair descriptions: *Change-of-Address* and *New-Patient-Assessment*. The initial ones are descriptions of service-request states of affair, while the final ones are descriptions of processed states of affair.

For instance, the New-Patient-Assessment initial state of affairs describes a situation where a *patient* speaks with a *case worker* of a *community care department*, and supplies to him/her the *new address* and *moving date* information. The case worker retrieves more *information about the patient* from the system, and then notify the new data. In this paragraph, italic words represent elicited concepts of the context that have been used to describe the viewpoint.

Note the absence of dynamics in the description. The case worker requires and supplies information, but we do no know when and how.

Interaction analysis

The *Interaction Description Module* represents an agreement between user and provider – or provider and provider – viewpoints about how to consume services and exchange resources. It is the unique point of contact: a shared module that represents knowledge crossing multiple viewpoints. It allows the capture of context elements and requirements that cannot be caught in other CLEO modules, but also check the existing ones. The core of the module is the *transition-event*. It gathers the elements that allow the representation of the involved agents, sequence of transitions, activation state, exchanged resources, and eventual data and process differences between two viewpoints. Because the heterogeneity of different viewpoints, we may expect at least two distinct counterparts of the involved descriptive entities: one (or more) from the domain ontology of the source viewpoint and one (or more) from the domain ontology of the destination viewpoint. This simple mechanism allows data and process mismatches between the shared elements of the two viewpoints. Since the limited space available, we cannot detail all of the aspects connected to the interaction module.

Based on such a module, the present analysis refines the existing descriptions, considering new aspects such as the dynamic of the scenario (i.e. the interaction between viewpoints), the source and the destination of the exchanged resources, the condition for exchanging resources, etc. This means that elements captured by a viewpoint can be introduced into other viewpoints. The constraints defined in the interaction module impose a rigour check in the definition of the new elements. In particular, they require that concepts playing a state and resource role in a transition should also play a role in a defined state of affair description.

In the patient moves house life event, we described five interactions between the three involved viewpoints: case worker, community care, and housing department. Figure 7 depicts the interaction descriptions linking the three viewpoints. The arrows indicates the direction of the transferred value: single-way arrows represent a communication interaction; double-way arrows represent a transactional interaction.

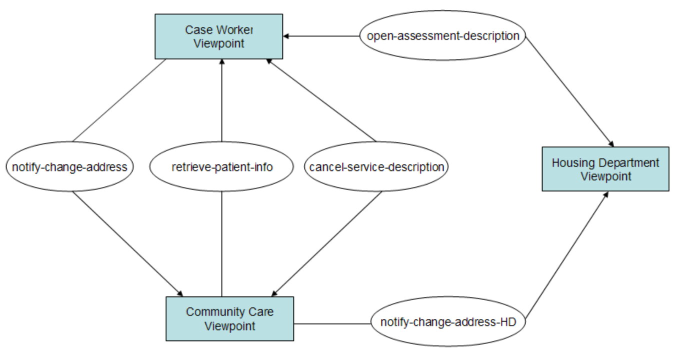

Figure 7: Identified interactions between the case worker, community care, and housing department viewpoints.

As an example, we consider the *Open-assessment-description*. It is a transaction that exchanges values between the case worker and the housing department in order to supply new care equipment to the patient and thus open an assessment. It describes two transition events that respectively represent: (i) a query of the case worker in order to obtain the *list of care equipments* that a patient can use; (ii) the opening of a *new patient assessment* based on of the available equipments. The two italic words represent the exchanged resources. The activation conditions of the two transitions impose the existence of specific parameters. In particular in the first transition, they require the existence of *patient-weigh* and *patient-impairment* attributes. Checking the constraints imposed by the module highlights that domain concepts selected by such descriptive entities do not play a role in the state of affair descriptions of the case worker viewpoint. This leads to the necessity of refining the New Patient Assessment initial state of affairs (that is the one provided as an example in the previous State of Affair analysis), introducing two new attributes that specialize the descriptive entities *patient-information*.

The check introduced a shortcoming in the state of affairs analysis. The bug regarded two possible service inputs, and thus the early discovery avoided problems at SWS definition level. For instance, developers could create a goal description using generic information of the patient as inputs and a possible web service that satisfy the goal using weight and impairment information as a goal. This problem could be solved only by introducing complex mediators between the goal and web service descriptions.

Both resource and activation elements select distinct concepts from the two distinct domains. This represents a data mismatching, that will solved later in the Service Ontology with the creation of appropriate OO-mediators.

Conception analysis

The conception analysis allows us to describe what an actor may conceive in a particular state of affairs. The conception description is the core of each viewpoint linking together all of the elements. Each viewpoint naturally focuses on different aspects of a life event: the user one includes the description of his/her needs; the provider one defines the offers; the manager one defines the policies that influence the service implementations; the politician one describes the laws that rule the scenario. The user and provider viewpoints represent the *configuration knowledge* of the scenario. The manager and politician viewpoints represent the *re-configuration knowledge*. The latter drives the evolution of the scenario. A change in an element of the reconfiguration knowledge level may produce changes in the other elements of the scenario. Changes can be propagated following the chain created by the *influences* relations that links the conception description elements.

In this case study, we focus on the configuration and service delivery knowledge levels. In the specific cases of user and provider viewpoints, the created need and offer descriptions link to one or more interaction descriptions. Goal and service descriptions represent the decomposition of the conception in active/computable steps that link to specific transitions of the associated interaction description.

A complex service is a service that allows to represent its functional decomposition into sub-services by means of a plan description. Sub-services may be known a-priori – in this case we can speak of **composition of services** – or their functionalities may be delegated to not known external services by means of a need description – in this case we can speak of **integration of services**. Actually, a service may be decomposed in terms of *service-description* or *need-description* concepts.

As an example we consider the definitions associated with the *open-assessment-need* defined by the case worker and the *open-assessment-offer* defined by the housing department. They both refer to the interaction *open-assessment-description* described in the previous phase. Figure 8 depicts the specific situation we are going to describe. Note that the interaction and conception description modules are tightly connected. The mechanism of need/goal and offer/service decomposition allows to model knowledge at different level of granularity, fitting project-specific requirements, and represent complex interactions that cannot be represented by the current one-shot SWS approaches.

open-assessment-need: the case worker needs to open a new assessment, after checking the list of care equipments that are eligible to a patient. It uses the following two goals: *list-equipments-goal*, and *open-assessment-goal*. These goals are respectively invocation for the transitions *list-equipments-event* and *open-assessment-event* (Fig-

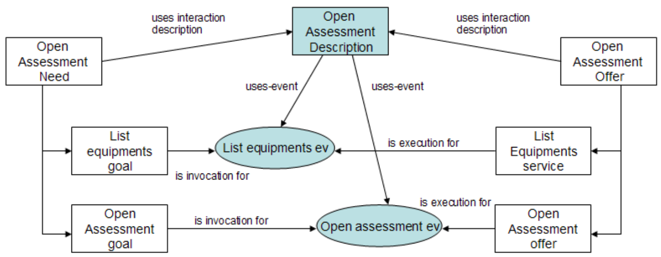

Figure 8: Links between Need and Offer Descriptions through the Interaction Description (gray boxes).

ure 8). The need description uses the plan *open-assessment-need-plan* for representing the sequence of the two goals.

open-assessment-offer: the housing department offers to open a new assessment after supplying the list of care equipments that are eligible to a patient. It uses the following two services: *list-equipments-service*, and *open-assessment-service* (Figure 8). The former is execution for the transition *list-equipments-event*; the latter is execution for the transition *open-assessment-event*.

The service *list-equipments-service* is a complex service that uses the need description *retrieve-list-equipments-need* for describing a **delegation** in terms of the following three goals: *finds-items-matching-weight-goal*, *finds-items-matching-impairment-goal*, and *list-intersection-goal*. The first requires a service that finds care equipments for a patient with a specific weight, the second one finds care equipments for a patient with a specific impairment, and the third one intersects the results of the previous two services. The need description specifies the plan *retrieve-list-equipments-need-plan* that arranges the three used goals.

Plan analysis

This is the last stage of the viewpoint analysis. Each viewpoint is completed with the description of all of the plans that describe procedures, processes, etc.

In our approach, we take advantage of a number of concepts from the *Ontology of Plan*, which is a module of *D&S* ontology. It allows the division of tasks into elementary and control and the construction of complex tasks from elementary ones among other features. In other words, we can describe both simple (e.g. workflow) and complex (e.g. scheduling) plans adapting to the needs and skills of the different actors: users, manager, organizations, etc. However, further specific approaches used by involved organizations may be adopted simply extending this module.

The plans organize goals or services within a need or offer description, or need and offer within a life event description (viewpoint). For instance, the plan for the needs of the case worker viewpoint describes the following sequence:

1. Get patient information;
2. Notify change of address;
3. Cancel services;
4. Open assessment.

The sequence represents the four steps that a case worker should follow in order to accomplish all the tasks connected to a patient moves house life event.

As further example, we consider the complex service introduced in the previous phase: *list-equipments-service*. It delegates its functionalities by means of a need description that contains three goals. Two of them ask for a list of equipments respectively on the basis of client weight and impairment. The third one asks for intersecting the two above lists and can be invoked only after the other two. The associated plan introduces the following couple of control tasks: *any-order-task* and *syncro-task*. All of the tasks within this couple of tasks can be executed freely. The syncro-task synchronizes the previous tasks before the execution of the last one. As a result, the plan can be represented as the following sequence:

1. **any-order-task**
2. finds-items-matching-weight-goal
3. finds-items-matching-impairment-goal
4. **syncro-task**
5. list-intersection-goal

Model Specific Scenario

Once the generic model has been created, to assess its viability it is tested with some specific scenario. We create instances of the concepts of the Domain Ontology (e.g. the Essex county council, the Chelmsford district council, dummy citizens and case workers, etc.). These instances populate the lexical layer of the ontological framework.

Further instances may specialize all of the descriptions of the context, describing specific cases. These instances are created starting from the lexical layer: we select instances of the Domain Ontology, we create instances of the descriptive entities that are played by such instances, and then we compose them into an instance of a context descriptions. This is a sort of revers path compared with the one we adopted for eliciting the knowledge. For example, an instance of a viewpoint description (e.g. the case worker viewpoint) is built selecting the instances playing a role in the description (e.g. the Jessica, Robert, Essex etc.), creating the instances of the roles of the description (e.g. jessica-patient, robert-case-worker, essex-county-council, etc.), and finally composing the situation following the defined relations (e.g. jessica-patient speaks with robert-case-worker, etc.). The result is a specific description of a scenario.

The creation of instances is a useful mean for checking the consistency of the created descriptions. Inferences based on the axiom and rules of CLEO can help in the creation of the instances (e.g. we can infer the elements of a transition starting from the defined elements of related state of affairs), as well as identify any lack in the model. This is the second and more accurate check point of the model (the first was the interaction analysis). In fact, we are able to test all the inference paths provided by the axiomatization of CLEO.

Create the SWS descriptions

The model created so far describes the context where the services are requested and provided and the involved concepts (Figure 9). The obtained descriptions may be the input for the creation of WSMO goals and web services.

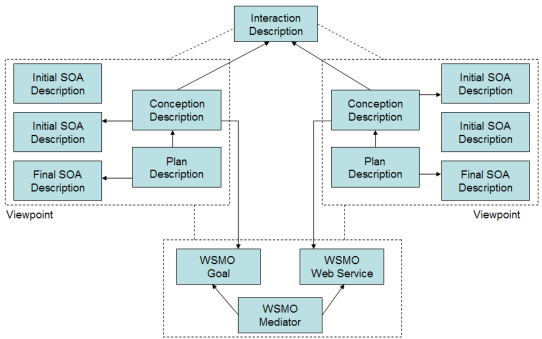

Figure 9: Creating the SWS descriptions.

As an example, we report the definitions of goal, web service, and mediator connected to the *open assessment* transaction between the *case worker* and *housing department* viewpoints.

The first step is to create the **WSMO-goal** description. The reference goal in the context is the *list-services-goal*, defined in the case worker viewpoint. Following the defined relations, it is possible to access to its associated state of affair descriptions (*New Patient Assessment*) and to the specific transition description it is invocation for (*list-equipments-resource*). The axiomatization allows to obtain the possible inputs and outputs of the WSMO goal simply inferring the description entities defined in the states of affair (initial for inputs, final for outputs) whose counterparts in the Domain Ontology are also counterparts of state or resource elements of the associated transition. The resulted inputs are *patient-weight* and *patient-impairment*, while the output is the list of *eligibility-equipments*.

In this case, the context does not provide any suggestion about specific capabilities for the goal. Neither the state of affair descriptions nor the transition state conditions defines specific constraints – except the existence of the inputs and the outputs, that is implicit with the WSMO goal input and output role definitions.

The second step is to create the **WSMO-web-service** description. The reference service in the context is the *list-services-service*, defined in the *housing-department* viewpoint. Using the same reasoning of the WSMO goal case, we can create the definitions of input and output. The inputs are the *client-weigh* and *client-impairment*, and the output is the list of *eligibility-equipments*.

Moreover, we introduce choreography and orchestration descriptions (interfaces). Each transition the service is execution for may be mapped to a choreograohy guarded transition. The set of all obtained guarded transitions is part of the choreography (other guarded transition can be added by developers for managing more detailed aspects; e.g. errors, acknowledge messages, etc.). In our example, the service description only links to the transition *list-equipments-event*. From such a transition, we can (i) derive the conditions of the guarded transition, referring to the transition condition *list-equipments-state-condition*, and (ii) define the call to a function that retrieves the transition resource element *list-of-equipments*.

The considered service is a complex service, and hence defines a functional decomposition. The orchestration is based on such a decomposition and can be defined in the format: (*Sequence* G1 G2 G3 M1 M2), where G1, G2 and G3 represent the goals and M1 and M2 the GG-mediators connecting them (Figure 10).

The last step is the creation of **WSMO mediator** descriptions. The existence of WG-mediators and OO-mediators is proved by means of axiomatization. The mediator descriptions used in this example (Figure 10) are explained in the following.

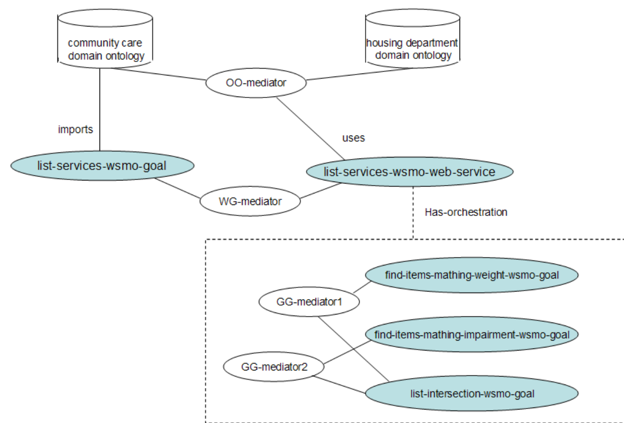

Figure 10: Sample structure of WSMO descriptions for the list service example

WG-mediator: connects *list-services-wsmo-web-service* to *list-services-wsmo-goal* allowing it to be selected for solving the goal. This mediator defines a mediation service for converting the value of input weight from pounds (in the goal) to kilos (in the web-service).

OO-mediator: Defines mapping rules for aligning *housing department* domain ontology (used by the Web Service) with *community care* ontology; for instance, it aligns the concept *impairment-HD* to the concept *impairment* that are used as input roles in the firs and the second ontology, respectively.

GG-mediator1: Allows the output of *find-items-matching-weight-wsmo-goal* to be used as input by *list-intersection-wsmo-goal*.

GG-mediator2: Allows the output of *find-items-matching-impairment-wsmo-goal* to be used as input by *list-intersection-wsmo-goal*.

Conclusion and Future Work

In this paper, we provide an ontological framework with which most PA's – or generally organizations – can identify, from which they can work when designing and delivering e-Government services. This general framework can be adapted and applied as appropriate. We present our requirements and approach in the construction of the conceptual model, and briefly describe the well-known ontologies that are the basis of our framework: WSMO and DOLCE.

The framework is composed by three ontologies – Core Life Event Ontology, Domain Ontology, and Service Ontology – that map a distributed e-Government service-supply scenario where multiple actors are independent nodes describing their own knowledge, and provide mechanisms of knowledge sharing and mismatch resolution.

Our approach allows the contextualization of the e-Government scenario in terms of descriptions. In particular, we introduce the following two separations: *Context vs Services* and *Context vs Vocabulary*. The former distinguishes between the description of the environment where the services are provided, used, and managed by different actors, and the description of the actual services that can be invoked. The latter distinguishes between descriptive entities of the context, and the actual objects of the actors' vocabulary they act upon.

As main result, we describe the Core Life Event Ontology (CLEO) that is the heart of our framework allowing the description of the configuration and re-configuration knowledge levels. It is a big ontology: contains 242 elements among classes, relations, axioms, and rules that compose its 6 modules. Moreover, it refers to specific modules (e.g. *Plan Ontology*) provided by the *D&S* ontology. For these reasons, we cannot detail all of the involved aspects in this paper.

We focus on the associated knowledge elicitation methodology that first helps domain experts to create a full description of a specific e-Government context using models close to their experience and specific languages of the different involved domains, and then drives the application developers to implement SWS descriptions inferring knowledge from the context description. To introduce the methodology and associated knowledge structures, we worked out a case study within the change of circumstances scenario.

Further important aspects, simply outlined in this paper, regard the description of the evolution of the scenario based on the politician and manager viewpoints, and the mechanism based on the need/offer and interaction module for describing complex transactions between viewpoints.

Finally, future work will concern the adoption of the ontological framework as the base of a middleware for service-oriented e-Government applications (e.g. Web Portals); i.e. creating the infrastructure for the semantic interoperability.

References

Cabral, L.; Domingue, J.; Gutierrez, L.; Rowlatt, M.; and Davies, R. 2005. WP 9: Case study egovernment D9.5 change of circumstance wsmo descriptions. *DIP Deliverable*.

Domingue, J.; Cabral, L.; Hakimpour, F.; Sell, D.; and Motta, E. 2004. IRS-III: A platform and infrastructure for creating wsmo-based semantic web services. In *proceedings of Workshop on WSMO implementation (WIW 2004), Frankfurt, Germany*.

Drumm, C.; Cabral, L.; Domingue, J.; Fuchs, H.; Gutierrez, L.; Rowlatt, M.; and Davies, R. 2005. WP 9: Case study egovernment D9.4 change of circumstances prototype. *DIP Deliverable*.

Dumitru, R.; Holger, L.; and Uwe, K. 2004. Web Service Modeling Ontology (WSMO). *Website: http://www.wsmo.org/2004/d2/*.

eGOV. 2004. eGOV project. *Website: http://www.egov-project.org*.

Engers, T. V.; Patries, J.; Kordelaar, J.; Hartog, J. D.; and Glasseee, E. 2004. E-Power project. *Website: http://lri.jur.uva.nl/epower*.

EU-PUBLI. 2004. EU-PUBLI project. *Website: http://www.eu-publi.com*.

Fernandez-Lopez, M. 2001. Metodo bi-fase para la conceptualization de ontologias basado en meta-modelos. In *PhD thesis*. Facultad de Informatica. Universitad Politecnica de Madrid. Spain.

Gangemi, A., and Mika, P. 2001. Understanding the semantic web through description and situations. In *proceedings of Confederated International Conference DOA, CoopIS, and ODBASE*. LNCS. Springer.

Gugliotta, A.; Cabral, L.; Domingue, J.; and Roberto, V. 2005. A semantic web service-based architecture for the interoperability of e-government services. In *proceedings of International Workshop on Web Information Systems Modeling (WISM 2005), Sydney, Australia*.

ICTE-PAN. 2004. ICTE-PAN project. *Website: http://www.eurodyn.com/icte-pan*.

OWL-S Coalition. 2004. OWL-S 1.1 release. *Website: http://www.daml.org/services/owl-s/1.1/*.

Oltramari, A.; Gangemi, A.; Guarino, N.; and Masolo, C. 2002. Sweeting ontologies with dolce. Ontologies and the semantic web. In *Proceedings 13th International Conference, EKAW 2002*. Siguenza, Spain.

OntoGov. 2004. OntoGov project. *Website: http://www.ontogov.com*.

SmartGov. 2004. SmartGov project. *Website: http://www.smartgov-project.org*.

Sycara, K.; Paoulucci, M.; Ankolekar, A.; and Srinivasan, N. 2003. Automated discovery, interaction and composition of semantic web services. *Journal of Web Semantic* 1(1).

T. Berners-Lee, J. Hendler, O. L. 2001. The semantic web. *Scientific American* 284(4).

TerreGov. 2004. TerreGov project. *Website: http://www.terregov.eupm.net/my_spip/index.php*.

van der Aalst, W.; ter Hofstede, A.; and Weske, M. 2003. Business process management: a survey. In *proceedings of International Conference on Business Process Management (BPM)*. pp. 1-12.