

Journal of Theoretical and Applied Electronic
Commerce Research

E-ISSN: 0718-1876

ncerpa@utalca.cl

Universidad de Talca
Chile

Brusa, Graciela; Caliusco, María Laura; Chiotti, Omar
Enabling Knowledge Sharing within e-Government Back-Office Through Ontological Engineering
Journal of Theoretical and Applied Electronic Commerce Research, vol. 2, núm. 1, april, 2007, pp. 33-
48
Universidad de Talca
Curicó, Chile

Available in: <http://www.redalyc.org/articulo.oa?id=96520104>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

Enabling Knowledge Sharing within e-Government Back-Office Through Ontological Engineering

Graciela Brusa¹, María Laura Caliusco² and Omar Chiotti³

¹ Dirección Provincial de Informática, Santa Fe, gbrusa@santafe.gov.ar

² UTN-CONICET, Fac. Regional Santa Fe, mlcaliusco@ceride.gov.ar

³ UTN-CONICET, Fac. Regional Santa Fe, chiotti@ceride.gov.ar

Received 02 October 2006; received in revised form 04 January 2007; accepted 28 February 2007

Abstract

Nowadays, organizational innovation constitutes the government challenges for providing better and more efficient services to citizens, enterprises or other public offices. E-government seems to be an excellent opportunity to work on this way. The applications that support front-end services delivered to users have to access information systems of multiple government areas. This is a significant problem for e-government back-office since multiple platforms and technologies coexist. Moreover, in the back-office there is a great volume of data that is implicit in the software applications that support administration activities.

In this context, the main requirement is to make available the data managed in the back-office for the e-government users in a fast and precise way, without misunderstanding. To this aim, it is necessary to provide an infrastructure that make explicit the knowledge stored in different government areas and deliver this knowledge to the users.

This paper presents an approach on how ontological engineering techniques can be applied to solving the problems of content discovery, aggregation, and sharing in the e-government back-office. This approach is constituted by a specific process to develop an ontology in the public sector and an ontology-based architecture. In order to present the process characteristics, a case study applied to a local government domain is analyzed. This domain is the budget and financial information of Santa Fe Province (Argentina).

Key words: Ontology, Public Sector, Budgetary and Financial System, Knowledge Sharing

1 Introduction

The term *e-Government* refers to the use of new technologies to transform public administration and to radically improve the services provided to citizens, enterprises or other public administration sectors. Nowadays, some initiatives related to e-government are focused on offering public services over the Internet through e-government portals without considering functional integration. Thus, to find the desired service, users must navigate over a huge number of websites that were designed using quite different criteria. Wherefore, it can occur either they do not find the service that might meet their needs or they must go over several links, which discourages their intention of solving their problem on-line.

Public administration must provide to users the correct information at the right time without making them waste their time. To this aim, organizational innovation through process redesigning and delivering interoperable services constitute the challenges that government must face. Then, providing integrated information is the key for delivering services that meet the community needs.

In the last years, information integration between heterogeneous information systems has attained significant improvements in the private sector. A replication of these advances to the public sector is questionable due to particular characteristics of public administration [33]. However, approaches proposed in this environment are based and grounded on private sector standards and initiatives, especially those referring to communications, libraries and business practices [13]. Then, adjustments of these advances to government organizations background must be made.

The problem of information integration brings about the emergence of the *interoperability* concept, its particular characteristics and its different aspects to consider: technological, organizational and semantic. The semantic aspect completes the information integration and exchange, because it provides the knowledge needed to appropriately use the data to be integrated. For this aspect, ontologies are becoming increasingly popular since they allow for delivering a shared common description of data that does not depend on the particular context of a data source, and can be freely communicated between information systems and people.

The objective of this paper is to present a specific process to develop an ontology in the public sector and to propose an ontology-based architecture to sharing knowledge within the back-office government. A case study applied to a local government domain is analyzed. This domain is the budget and financial information of Santa Fe Province (Argentina).

The paper is organized as follows. Section 2 describes the integration needs of e-government, interoperability and ontological engineering. Section 3 presents a process to develop an ontology in the public sector. Section 4 presents an ontology development using the mentioned process and its implementation using Protégé 3.1. Section 5 introduces the architecture to support information integration using the implemented ontology. Finally, Section 6 presents conclusions and future work.

2 E-government Interoperability and Ontological Engineering

Government administration includes a wide range of tasks. Some tasks are structured and completely supported by information systems and others are semi-structured, and then depending on the professional knowledge [22]. But the information is always a critical resource that must be available and accessible in a complete, fast, safe and reliable fashion because governments must give solution to the wide variety and amount of current social requirements. For that reason, e-government requires innovating processes to satisfy this information requirement. This brings about an opportunity to work on the integration aspects but, on the other hand, it constitutes a great challenge.

The characteristics of the government structure and its network of public services force to consider the integration problem from many viewpoints. The holistic reference model proposed by [38] shows different perspectives that must be taken into account in government work plans to provide services to citizens. From these perspectives, e-government activities can be concentrated on two well differentiated work fronts: an external front or front-end, usually presented through government websites, which integrates public services according to citizens requirements and interests; and an internal front or back-end for communicating and integrating the information dealt by the State in performing its administrative tasks and that constitutes the support and source of the information presented by those websites. At this place, back-office integrates information systems that support the organizations as human resources, accounting or budget and front-office gathers productive information systems as Health, Education, Social Action, Justice, among others.

It is in the back-office where the concept of e-government is consolidated. Tasks in the administration itself are quite hard and they are usually unknown to community, and they are responsible for the quality of the provided services. Then, the heterogeneity problem arises as the core aspects to be solved.

Heterogeneity can be classified into three types: technological, organizational, and informational. Technological heterogeneity refers to the existing technical diversity, such as platforms, protocols, equipment, work environments and methodologies, among others. Organizational heterogeneity refers to the elements with different features that must be taken into account to solve problems in the State, such as processes, work criteria, actors, decision and guidelines levels, among others. Finally, information heterogeneity is related to two kinds of situations [36]:

1. Structural heterogeneity, which arises when data are kept in different data structures.
2. Semantic heterogeneity, when different data have the same meaning or when a unique data refers two different concepts.

In this context, interoperability has emerged almost naturally and, in fact, it is an issue that is being approached from different fields.

2.1 Interoperability

In the literature, there are different definitions about interoperability [3]. This paper considers the definition offered by [30] and [26]. The authors state that interoperability refers to the process of ensuring that information systems, procedures and culture of an organization are managed with the aim to maximize opportunities for the exchange and re-use of information, whether internally or externally. In the context of the State reform, this implies working inside the administration both on its organizational aspects and on its attitude towards information exchange, intending to attain data integration, interaction and applications reuse, with the object of making decisions more efficiently and providing the answers required by citizens. It also means working on rules issues that set up clear and precise rules to lead different aspects involved in interoperability, such as technical, human, proceeding, exchange and security.

A simple way of classifying interoperability is the following:

1. Technical Interoperability. It covers the technical issues of linking computer systems and services. It includes open interfaces, interconnection services, data integration and middleware, data presentation and exchange, accessibility and security services. This is the first step to settle communication among parties. At this level, it can be mentioned standards, as TCP/IP and XML (eXtended Markup Language).
2. Organizational Interoperability. It refers to institutional and human issues required to arrive to agreements, consensus, cooperation and collaboration that overcome either any barriers inside the organization, or local, regional, national and international barriers. This leads to the development of legal aspects that guarantee access to resources, authentication, data protection issues, author rights and privacy protection, among others.
3. Semantic Interoperability. It refers to having the necessary knowledge so that computer services can understand the terminology, capacities and intentions of the communicating parties. In this context, there are taxonomies, thesaurus and ontologies [15]. There are also metadata initiatives, such as Dublin Core, which is widely adopted in the public sector.

Achieving interoperability in an organization is not a static process but a permanent task since once a perspective is solved; the problem is presented in another level. This paper is focused on solving the semantic interoperability in back-office to contribute for integration those e-government needs. In the context of semantic interoperability, the contribution of ontologies is quite significant because they provide a set of related concepts through an explicit model of relations, defining a particular phenomenon of reality with the consensus of the involved parties.

2.2 Ontological Engineering

There are different definitions of ontology since it has been used for different purposes in different disciplines. In [15] an analysis of those definitions is presented. The authors arrive to the following definition "ontologies are intended to capture knowledge that is formally and generically agreed and that can be reused and shared through applications (software) and by groups of people. Ontologies are usually cooperatively built by people that are generally located in different places".

This paper considers an ontology from a pragmatic point of view as, *a vocabulary of terms associated to a particular domain and specifications of their meaning through axioms and properties using a logic-based representation language*.

The main ontology components are:

1. Concepts: basic ideas of domain to formalize.
2. Relations: associations between concepts of the domain. Particularly, the relation *Subclass-of* or *is-a* is used for building the class taxonomy.
3. Properties: concept's characteristics (attributes).
4. Axioms: sentences that are always true in the domain of discourse. They are normally used to represent knowledge that cannot be formally defined by the other components. In addition, formal axioms are used to verify the consistency of the ontology itself or the consistency of the knowledge stored in a knowledge base. Formal axioms are very useful to infer new knowledge. Axioms can be divided into properties restrictions (value restrictions) and relation restrictions (inverse, functional).
5. Instances: they represent specifics objects of a concept.

Taking into account the ontology components, [24] proposes a lineal categorization shown in Figure 1. This categorization illustrates the idea that exists different structures depending on the components they define. However, when we talk about ontology all of these components have to be present.

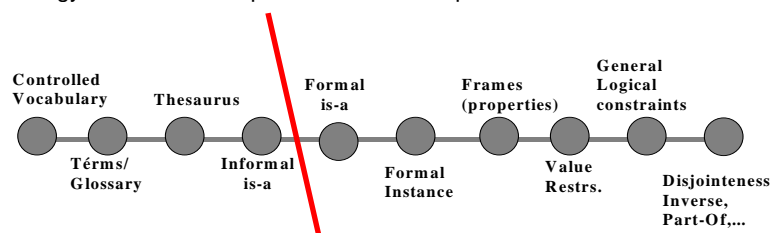


Figure 1: Lassila & McGuinness categorization [24]

With regards to methodologies, several proposals for building ontologies were defined [36] [7]. Two different groups of methodology can be figured out. The group of experience-based methodologies, represented by [16] a methodology defined from TOVE Project and, Uschold and King (1996) methodology [35] based on the experience of developing the Enterprise Ontology. The group of methodologies that define a set of activities to develop ontologies based on its life cycle and a prototype refinement, as METHONTOLOGY [15] and, 101 Method [27]. Usually, the experience-based methodologies ones are appropriated for building ontologies when their purposes and requirements are clear; the methodologies of the second group are useful when the environment is dynamic and difficult to understand and the objectives are not clear from the beginning [9]. Moreover, it is common to merge different methodologies since each of them provides design ideas that distinguish it from the others. Mainly, this merging depends on the application that the developer has in mind, the tool that she/he uses to develop the ontology and the knowledge background she/he has.

According to [35], there is not a unified methodology for developing ontologies but different approaches for specific circumstances exist. In [11] the authors encourage this position saying that a technical area has reached its maturity when it takes into account largely accepted methodologies. He makes a comparison between better methodologies of Ontological Engineering against IEEE Standard for Developing Software Life Cycle Processes, 1074-1995 [20]. Taking into account software definition from IEEE, Fernández López [11] considers that ontologies are part of software products and then proposes standard processes from Software Engineering for ontologies development.

2.3 Related work

In the most so-called advanced nations, there are several initiatives that face the problem of providing e-Government services [1]. The most important initiatives are: SAGA in Germany, e-GIF in United Kingdom, ADEA in France, and FEAF in USA. Most of these proposals just provide some general recommendations for the development of software product. In developed countries, we can also find similar projects. Further there are a number of European Union projects e.g. Terregov [5] and OntoGov [2] that make use of semantic technologies for achieving interoperability and integration between e-Government systems.

Particularly, in Argentina, an organism responsible for implementing a national plan of electronic government has been created. This organism, called ONTI (National Office of Information Technology), has been created a portal for the Argentinean government. This portal is available from www.argentina.gov.ar.

The main goal of the projects previously discussed, is to allow local government and government-related agencies to offer online access to their services in an interoperable way. Then, they are focus on providing semantic interoperability between the front-office activities. In contrast to these projects, we are interesting on providing an infrastructure for knowledge sharing in the back-office.

3 Ontology Development Process

The objective of this section is to present a development process for building ontologies in public sector. This process was developed following two main objectives: to allow a more adequate representation for a complex local domain, and to allow an effective development of domain ontology from scratch. To this aim, the adoption of an evolutive prototype methodology, such as METHONTOLOGY, seems to be the more adequate decision. However, tools proposed by this methodology are insufficient for managing the great volume of data during the specification and conceptualization phases. Then, taking to this methodology as it bases, tools of knowledge representation and modeling were added. These tools were taken from other methodologies such as the Ontology Development 101 Method [27], the Uschold and King Methodology [35] and, Grüninger and Fox Methodology [16]. Finally, some intermediate representations based on mature software engineering techniques were included to allow the communication between ontology engineers and domain experts.

The ontology development process is presented in Figure 2. The process is divided into three main subprocesses: Specification, Conceptualization and Implementation.

The objective of the specification subprocess is to acquire informal knowledge about the domain. To fulfill this objective this subprocess is divided into four main tasks: determine ontology goal and scope, describe the domain, define motivating scenarios and competency questions and, define granularity and ontology type. With regards to the conceptualization subprocess, its objective is to define a domain conceptual model organizing the relevant knowledge acquired in the previous subprocess. To this aim, this subprocess is divided into three main tasks: define the domain conceptual model, identify classes, relations and attributes and, create instances. Finally, the goal of the implementation subprocess is to build a correct ontology represented in a machine-processable language. With this goal in mind, this subprocess is divided into three main tasks: implement the ontology, verify the ontology and, validate competency questions.

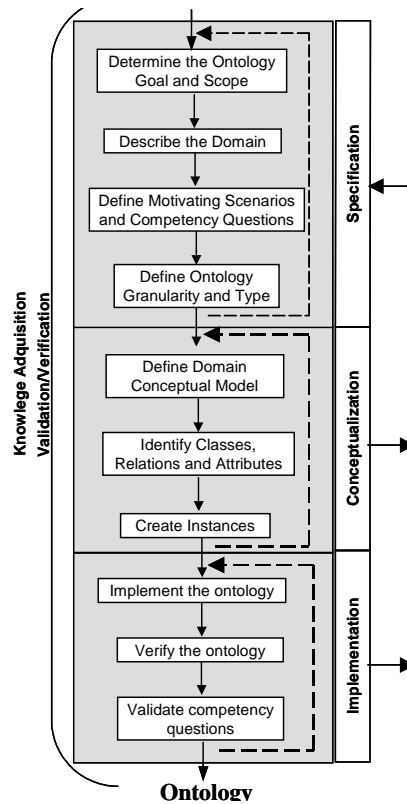


Figure 2: An Ontology Development Process

This process proposes an iterative development of an ontology and, during the iterations the knowledge acquisition activities and, validation and verification activities. In the context of software engineering, formal verification is the act of proving or disproving the correctness of a system with regards to a certain formal specification or property, using formal methods. To prove the correctness of the ontology, consistency, completeness and conciseness have to be verified. Ontology validation refers to whether the ontology definitions really model the real world for which the ontology was created [15]. Then, validate the ontology means to check if the ontology fulfill the requirements defined in the ontology requirements specifications.

4 Building a Domain Ontology from Scratch

The objective of this section is to show an application of the process defined in the previous section. Particularly, it was applied to develop an ontology that describes the semantics of the government budgetary domain in Santa Fe Province, Argentina. The budget is an information system with a critical mission. Its information crosses the whole government structure and, the state reform tasks in Argentina have been initiated through this system. Local budget life cycle (Figure 3) is complex because it involves a sequence of different instances with a lot of data; and a specific knowledge is required to operate with them. Along this life cycle the evaluation and control of actual and financial resources is made, and all of them are assigned to good and services production.

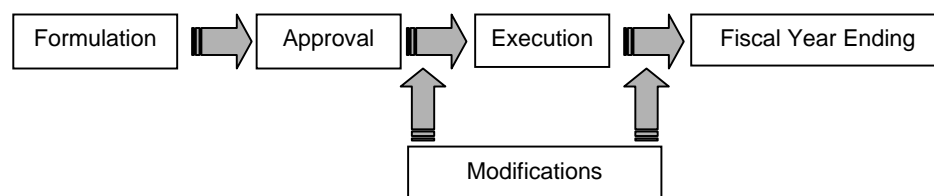


Figure 3: Stages of the Local Budget Life Cycle.

The availability of knowledge associated to budgetary data is always critical in each stages of the local budget life cycle. In formulation, execution, modifications and fiscal year ending stages, only government staff with specific knowledge can be involved in each other, concentrating a great responsibility on few people groups. The communication among them is difficult because they have a portion of common vocabulary and specifics concepts to each other that they do not share. In the approval stage, semantics information it is necessary for analyzing

budgetary data and then having the budget law passed. Here, it is more complex because all the legislators must vote and most of them have no specific knowledge. For simplicity, the formulation stage for expenses budget was considered for this study case.

4.1 Iteration I

4.1.1 Specification: Goal and Scope of the Ontology

The scope limits the ontology, specifying what must be included and what must not. In 101 Method, this task is proposed in a later step but in this work it has been considered appropriate to include it at this point for minimizing the amount of data and concepts to be analyzed, especially due to the extent and complexity of the budgetary semantic. In successive iterations for verification process, it will be adjusted if necessary.

The objective is to develop an ontology that describes the semantics of the budget formulation task. This ontology only considers the needs to elaborate a project of budget law with concepts related to expenses associated to the formulation stage. It does not consider the concepts related to other stages as budgetary executing, accounting, payments, purchases or fiscal year ending.

4.1.2 Specification: Domain Description

Taking into account that this work was made from scratch and that 101 Method proposes the enumeration of main terms to continue as well as METHONTOLOGY plans using intermediate representations for organize knowledge domain in the conceptualization phase [15], it was necessary to make a previous domain analysis. In this analysis, the application to formulate budget and related documentations were studied and revised, meetings with experts were carried out, and informal documentation and information were compiled.

The budget of a government is a plan of the intended revenues and expenditures of that government. The budget is prepared by different entities in different government areas. Particularly, in the Santa Fe Province (Argentina) these entities are:

1. *Executive Power*: this government entity, formed by a Ruling Organism and Executing Organisms, elaborates the Provincial Budget Draft. The first organism defines all activities for formulating a budget and the others execute these activities.
2. *Legislative Power*: it passes the Annual Budget Law.

Interaction among these actors leads different budget states: In Formulation, in Parliamentary Proceeding and Approved. This iterative process is shown in Figure 4.

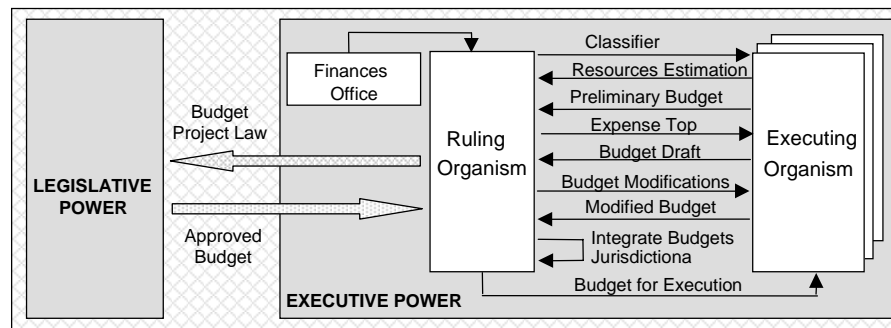


Figure 4: Iterative process until budget is ready for execution

In the Executive Power, a Ruling Organism is responsible for the budgetary formulation process. This Ruling Organism sets the budgetary policies and drives the jurisdictional interactions to complete and integrate its own expenses and resources estimated through this formulation process. Each jurisdiction as Health or Financial Ministries has an Executing Organism, which is in charge of formulating and executing the budget. Formulation process results in the Budgetary Project Law issued to Legislative Power for approval. Once the Budget Law is passed, there are two possible states for budget: In execution and Closed.

4.1.3 Specification: Motivating Scenarios and Competency Questions

This step has been included taking into account the opinion of [16]. The authors consider that for modeling ontologies, it is necessary an informal logic knowledge model in addition to requirements resulting from different motivation scenarios. The motivation scenarios show problems that arise when people needs information that the information system does not provide. Besides, the scenarios description contains a set of solutions to these problems in which the semantic aspects to resolve them are.

Particularly in this work, in order to define motivating scenarios and communicate them to involved people, templates have been used. These templates were based on those proposed to specify case uses in object-oriented methodology [34] and it was included in this work to reach severity to representation. The template describing the scenario of Local Budget Formulation is presented in Table 1. In this table are described: the name of the scenario, a brief scenario description, the location where the scenario occurs, the public agents who participate in the scenario,

a set of requirements that must always be fulfilled prior to the execution of the scenario, a list of needs required to execute the scenario, a set of tasks that define the normal sequence of the scenario, a condition that must always be true just after the execution of normal sequence of the scenario, actions that are not part of normal operations or standards, a list of possible problems caused by semantic heterogeneity and a list of possible terms related to the scenario.

For the public sector, characterized by a lot of sceneries and great quantity of concepts, this representation form helps to organize the domain knowledge. It allows to structure main common aspects facilitating both to carry out this task with users and to compare scenarios. Besides, template description can be extended.

SCENARIO N°	1	
NAME	Local Budget Formulation	
DESCRIPTION	Necessary tasks to estimate expenses for next year, which will be integrate with the other government jurisdictions for elaborating Draft Local Budget.	
SITE	Executing Organism of a Jurisdiction	
ACTORS	<ul style="list-style-type: none"> Public agents uncharged jurisdictional budget Ruling Organism agents Public agents from areas of a jurisdiction 	
PRE-REQUIREMENTS	<ul style="list-style-type: none"> Budgetary Policy defined Expenses Classifiers received from Ruling Organism Reference documentation 	
ASSOCIATES REQUIREMENTS	<ul style="list-style-type: none"> Prepared agents in Budget Formulation tasks. Advisory agents from Ruling Organism 	
NORMAL SEQUENCE	STEP	ACTION
	1	To receive expenses estimations from jurisdiction areas
	2	To bring support to this areas for elaborating own expenses programs.
	3	To integrate all expenses programs for jurisdiction.
	4	To create Programming Categories and send it to Ruling Organism
	5	To create the Jurisdictional Budget Project
	6	To load budget in informatics system and send it to Ruling Organism
	7	To receive approved jurisdictional budget from Ruling Organism
POST-CONDITION	<ul style="list-style-type: none"> Jurisdictional Expenses Budget Project Jurisdictional Programmatic categories 	
EXCEPTIONS	STEP	ACTION
	5	To consult the Ruling Organism if it does not understand different aspects to formulate budget.
	7	To modify budget if it is not approved
MAIN PROBLEMS	<ul style="list-style-type: none"> A lot of time lost in clarifying conceptual doubts Great problems when an agent must be replaced in key places of work. The whole process is highly dependent of few people's knowledge. 	
MAIN TERMS	Budgetary classifier, expense a classifier, Institutional, Programmatic Category, Geographic, Expenses Object, Financing Source and Finality Function Classifiers, among others, for working into the budget draft task.	

Table 1: Scenario description

The competency questions proceed from motivation scenarios and they are the questions that users ask when they work with domain concepts. This allows deciding the ontology scope to verify if it contains enough information to answer these questions and to specify the detail level required for the responses.

Moreover, the competency questions allow defining a hierarchy so that an answer to a question may also reply to others with more general scope by means of composition and decomposition processes. Table 2 shows some of them.

Simple Questions	Complex Questions
Which are budgetary classifiers?	What are sector and subsector for Main Administration?
Which are expenses/resources classifiers?	What is the character code for "Decentralized Organism"?
Which are the executor organisms for Health Minister?	Which is the institutional code for "Pharmacological Producer Laboratory" SAF?

Table 2: Samples of Competency Questions

The use of templates for representing scenarios was very useful in practice for work with public agents. They bring an agile mechanism to structure the main aspects description avoiding oversights. This is especially appropriate in complex domains as public sector is. In the same sense, competency questions allow to understand how a worker thinks when executing a task and which are he/his semantic necessities. Furthermore, these questions help to determine the granularity level that the ontology will has.

4.1.4 Specification: Ontology Granularity and Type

According to the level of granularity defined in [15], the ontology proposed here has fine granularity because describes the vocabulary needed to develop a specific task as budget formulation is. It comprises general concepts of domain and specific terms for budget formulation task. Therefore, according to the classification of ontology types proposed in [17], it is an ontology task.

4.1.5 Conceptualization: Conceptual Domain Model Determination

In this step, a list of main terms was elaborated according to the 101 Method guide. To this aim, the middle-out-strategy [35] was used to define a key term list. This list, shown in Table 3, does not include partial or total overlapping of concepts, synonyms, properties, relations and attributes.

Activity	Budgetary Fiscal Year	Financial Administration	Program
Budget	Budgetary Policy	Financing Source	Subprogram
Budget Analytic	Budgetary Top	Geographic Locate	Program Executer Unit (UEP)
Budget Approved	Executing Organism	Institutional	Programmatic Category Project
Budget Project Draft	Expense	Institution	Project
Budget Synthetic	Expense Classifier	Financial Administrative Service (SAF)	Public Funds Administrative Service (SAFOP)
Budget States	Expense Object	Jurisdiction	Ruling Organism
Budgetary Classifier	Finality Function	Jurisdiction Government	Resource

Table 3: Key Terms List

After defining the key term list, a class hierarchy has to be elaborated. To this aim, we propose to represent the class hierarchy by using a UML [34] class diagram, defining a class called "Thing" as the top class of the hierarchy. If the diagram became complex and difficult to manage, it could be modularized by using packages.

Although UML in its standard form is not suitable for semantic representation, the use of graphical representation is suitable in order to facilitate the communication between ontology engineers and domain experts [10]. The UML class diagram can be used to express concepts in term of classes and relationships among them [8]. In addition, if an ontology-based application is being constructed using object-oriented technology, it may be advantageous to use the same paradigm for modeling ontologies and knowledge [8]. In the last years, some MOF-based ontology modeling languages were defined [6], however, there is not yet appropriate tools to use them.

The class hierarchy was the basis for building the ontology term glossary, trying to include other concepts by means of generalization and specialization techniques. The conflictive assertions over the same entity may be discovered if the concepts are described as completely as possible [21]. For this purpose, we made definitions as complete as possible to contribute to define rules and axioms.

This UML model was very huge but it was useful to take an important design decision: working with two ontologies. One of them contains the common concepts for the budget life cycle and the other, contains the semantic specific for formulating it. Working with different ontologies allows applying reusability and usability attributes.

This decision was made after noting that the information, which supports the budget formulation task, also contemplates support for other subprocesses of the budget life cycle. Besides, the model performed in UML allowed us to clearly identify the general concepts with reusing possibilities for other subsystems different from those that are particular to each task. Considering the ontology types classification in [17], it was considered that the most suitable step for the proposed scenario was the creation of a domain ontology with the most general terms of the formulation subprocess and that they are also used in other subprocesses of the budget life cycle and another ontology, including just the necessary terms for the specific formulation task.

In this way, the decision was to create first the domain ontology with the general concepts for the whole budget life cycle. In future, it would allow reusing by task or budget specific ontologies. This decision leads to an ontological model as the one proposed in [36] in Figure 5.c, in which the shared vocabulary is implemented with a domain ontology and the local ontologies are corresponded to the task ontologies for each budget subprocesses.

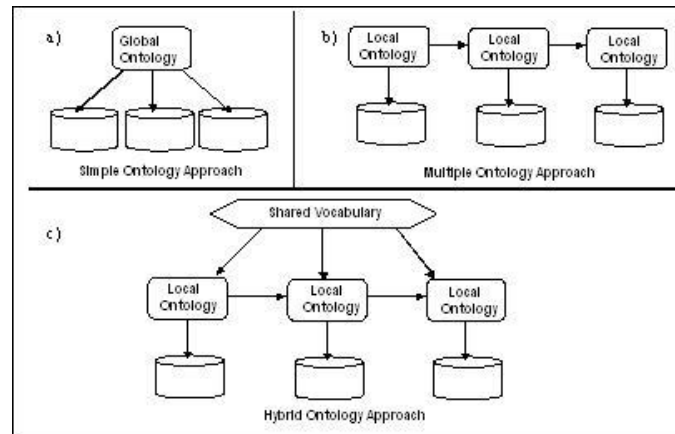


Figure 5: Ontologies Construction Forms [36]

For that reason, from now onwards, the development process of the ontology is started again (Figure 2) but for the domain ontology, leaving aside for a future work the creation of the budget formulation specific ontology, which will definitely have to use and relate to the domain ontology.

4.2 Iteration II

4.2.1 Specification: Goal and Scope of the Ontology

Now, this iteration goal is to construct a Domain Ontology oriented to work with general concepts for the budget life cycle. Because this ontology can be used in all budget stages, it enables reusability attribute. And, the ontology objective is to facilitate the communication between central administration staff that must deal with the local budget, bringing adequate terminology to non-expert- users, whether internally or externally.

The scope is stated by general concepts needed for formulation stage but common to other stages. It is out of this work scope these terms of own use for formulation task and another budget life cycle subprocess.

4.2.2 Specification: Domain Description

To achieve the goals of programming, assessing, and public incomes and expenditures control, the budget must be precisely specified. This is carried out by specifying its main classifying forms, such as income sources, expenditure characteristics, and incomes and expenses allocation in government institutions, according to its geographical distribution and also according to the programs planned for each of them. These different forms of specification determine a set of classifications to describe and expose the information coming from expenses and resources transactions made by public institutions.

These classifications are common information for the whole budget life cycle. Each of them depends on a specific purpose and they are called Analytical or Primary Classifications. Besides, there are relations among them that give rise to another type of classifications, which is the result of a primary information processing and they are called added budgetary classifications. Primary budgetary classifications are: Institutional, Geographical Location, Expenditure Object, Financing Source, Function Purpose, Programmatic Category and Resources Item.

4.2.3 Specification: Motivating Scenarios and Competency Questions

Domain Ontologies do not generally arise from motivation scenarios, since these scenarios describe the semantics of terms that are commonly used in a specific situation and, thus, shared by the different domain tasks. But, as in this case, a domain ontology can be defining by applying a generalization process of one or several particular scenarios. In this case, an exclusive scenario of the budget formulation task led to the creation of a domain Ontology.

Competence questions that can be considered in this ontology are those taken from Table 2 that refers to general aspects that have to be known for the specific formulation task. This includes all those questions that are related to the finished knowledge of budgetary classifiers and their use for the budget.

4.2.4 Specification: Ontology Granularity and Type

In this case study, the ontology is a formal structure expressed in formally defined languages for general terms of the budgetary domain of the Santa Fe Province. Its granularity is thin so that it can be reused by different government areas or in non-governmental areas. In the balance between specificity and availability, domain ontologies are created with a lower specificity to provide a higher availability, favoring reuse. For that reason, they are also called *online* ontologies [17].

4.2.5 Conceptualization: Conceptual Domain Model Determination

According with the description presented in Section 4.1.5, a new key terms list with most general terms was elaborated taking away specific terms used in the budget formulation stage, a glossary was reformulated and conceptual aspects in the context was modeled in a UML diagram with the main relations among these terms.

4.2.6 Conceptualization: Identification of Classes, Properties and Restrictions

At this step, we considered 101 Method guide and recommendations. Besides, we used representations proposed by METHONTOLOGY to knowledge organization as concepts classifier trees (Figure 6) to analyze hierarchies and attributes, binary relations, axioms and instances tables. For determining classes, we identified those terms of independent existence from the key terms list and the glossary.

Besides, disjoint classes, exhaustive decompositions and partitions [19] may be identified in these graphic representations.

1. A Disjoint-Decomposition- of a concept C is a set of subclasses of C that do not have common instances and do not cover C, that is, there can be instances of the concept C that are not instances of any of the concepts in the decomposition. As an example (see Fig. 6), Finality Function, Financing Source, Expense Object, Programmatic Category, Geographic Locate and Institutional can be mentioned as disjoint.
2. An Exhaustive-Decomposition- of a concept C is a set of subclasses of C that cover C and may have common instances and subclasses, that is, there cannot be instances of the concept C that are not instances of at least one of the concepts in the decomposition. For example (see Figure 6), the concepts Expenses Classifier and Resource Classifier make up an exhaustive decomposition of the concept Budgetary Classifier because there are no classifiers that are not instances of at least one of those concepts, and those concepts can have common instances.
3. A Partition of a concept C is a set of subclasses of C that do not share common instances and that cover C, that is, there are not instances of C that are not instances of one of the concepts in the partition. In this scenario there are no partitions.

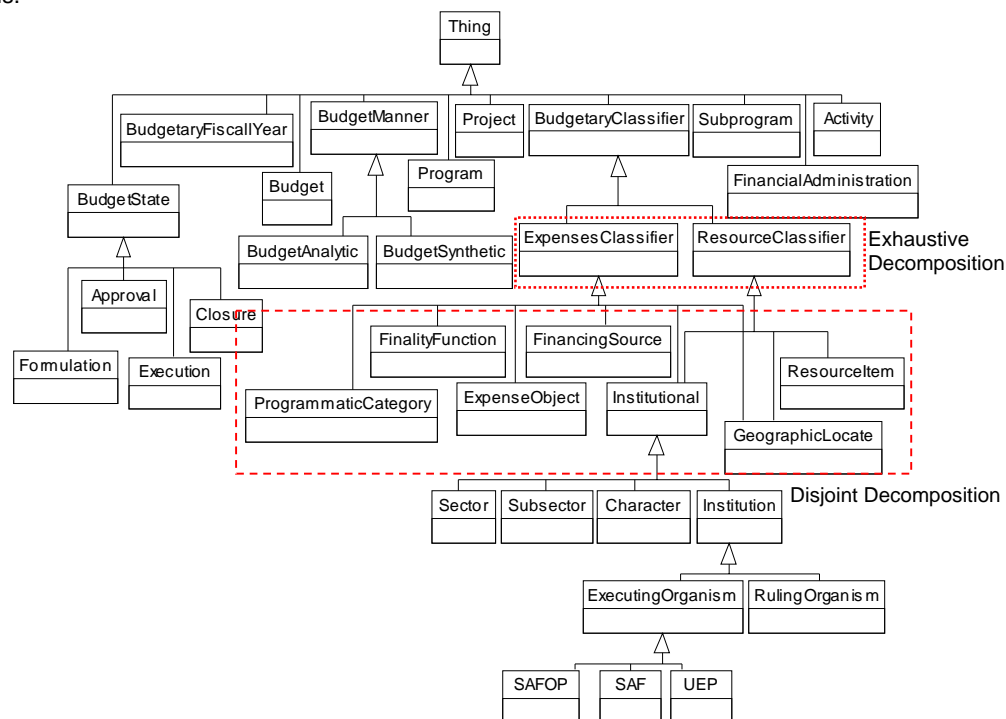


Figure 6: An excerpt of the Budgetary Ontology Taxonomy

Once the hierarchies and their features have been identified, a table to reflect bidirectional relations may be elaborated assigning names using a uniform criterion, identifying domain and range, cardinality and inverse relations. An example is shown in Table 4. This table can be used to identify, compare and verify restrictions and rules that ontological components are subjected to.

On the one hand, a restriction, in general usage, is a specific type of rule, which defines a finite (and generally absolute) boundary defined for a type of process or function. As regards ontology, restriction refers to constraints imposed by the way concepts are structured. For example, cardinalities and allowed values express restrictions. The restrictions can be assumed to be captured by the graphical notation, and should not be explicitly written [10]. Then, restrictions could be defined while defining class relations, attributes and properties.

On the other hand, a rule is a widely accepted norm, concept, truth, definition, or qualification in the domain of discourse. In public administration domain, the rules are clearly defined in the legal norms. Then, in order to define rules, the ontological engineer has not only to analyze the scenario description templates associated with the term

under consideration, but also the legal norms associated with these scenarios. Rules need to be explicitly defined using a formal language, such first order logics. This distinction is important to guide the ontological engineering on defining ontology constraints.

Concept	Relation	Cardinality	Concept	Inverse Relation
Institutional	inst-include-sec	1	Sector	sec-isPartOf-Inst
Institutional	inst-include-sbsec	1	Subsector	sbsec-isPartOf-Inst
Institutional	inst-include-char	1	Character	char-isPartOf-Inst
Sector	sec-isPartOf-Inst	1,n	Institutional	inst-include-sec
Subsector	sbsec-isPartOf-Inst	1,n	Institutional	inst-include-sbsec
Character	char-isPartOf-Inst	1,n	Institutional	inst-include-char
Institution	ins-has-SAF	1	SAF	SAF-correspond-inst

Table 4: An excerpt of the Bidirectional Relations Table for Budgetary Ontology

Then, taking into account the scenario description templates and legal norms, several rules were defined and written in natural language. Following, these rules were formalized in Description Logics. For example, the following axiom restricts that a ProgramaticCategory is an ExpensesClassifier and if it has a Subprogram it has to be associated to a Program.

$$ProgramaticCategory \subset ExpensesClassifier \cup \forall hasSubprogram. (\exists hasProgram. Program)$$

Finally, we have analyzed the axioms both individually and in a group of classes to verify if closure restrictions are required. Closure restrictions are a statement of a universal restriction that means a property only can be satisfied by a conditions exhaustive list.

4.2.7 Conceptualization: Instances Definition

The last step of conceptualization phase is to create individual instances of classes. In order to create individual instances of classes it is useful to analyze the scenario description templates and competency questions.

Defining an individual instance of a class requires (1) choosing a class, (2) creating an individual instance of that class, and (3) filling in the attribute values. According to METHONTOLOGY, each instance should be defined, as Table 5 shown, by its name, the name of the concept it belongs to, and its attribute values, if known.

Concept Name	Instance Name	Property Name	Value
Institutional	Institutional_111	cod-institutional	1.1.1
		has-fiscal-year	2004
		inst-include-sec	1-No Financial Local Public Sector
		inst-include-sbsec	1- Local Administration
		inst-include-char	1- Main Administration
	Institutional_212	cod-institutional	2.1.2
		has-fiscal-year	2004
		inst-include-sec	2-Financial Local Public Sector
		inst-include-sbsec	1-Official Banking System
		inst-include-char	2- Official Banks

Table 5: An excerpt of the Instance Table for Budgetary Ontology

4.2.8 Implementation: Construction of the Domain Budgetary Ontology

Implementing an ontology means to represent a correct ontology into a machine-processable language. There are several languages for implementing an ontology, however, the most relevant ones are RDF (Resource Description Framework) [4] and OWL (Web Ontology Language) [31].

The first challenge during the implementation phase is how to transform the UML class diagram (conceptualization phase) into a machine-processable ontology language. This task implies to transform composition relations into bidirectional relations. Some concepts modeled as classes in UML could be properties in ontology. Not all relations in UML have to be modeled in ontology but only those necessary to answer competency questions.

In order to carry out this activity, an ontology development tool could be used. In the last years, a new generation of ontology engineering environments has been developed. Among these environments, it can be mentioned Protégé 3.1 [14], WebODE [7] and OntoEdit [32].

The budgetary ontology has been implemented using Protégé 3.1 because it is extensible, and provides a plug-and-play environment that allows rapid prototyping and application development. Protégé enables exporting the ontology into RDF Schema (RDFS) and OWL. Particularly, the budgetary ontology has been implemented in OWL language.

To compare the ontology implementation with its conceptualization, graphical representations using the OWLViz and Ontoviz plug-ins were generated and compared with UML diagrams. On the one hand, OWLViz enables the class

hierarchies in OWL Ontology to be viewed, allowing comparison of the asserted class hierarchy and the inferred class hierarchy.

On the other hand, OntoViz generates graphics with all relations defined in the ontology, instances and attributes. As example, Figure 7 shows the main relations of the concept *Institutional* with other concepts, and an instance of this concept, *Local Administration*.

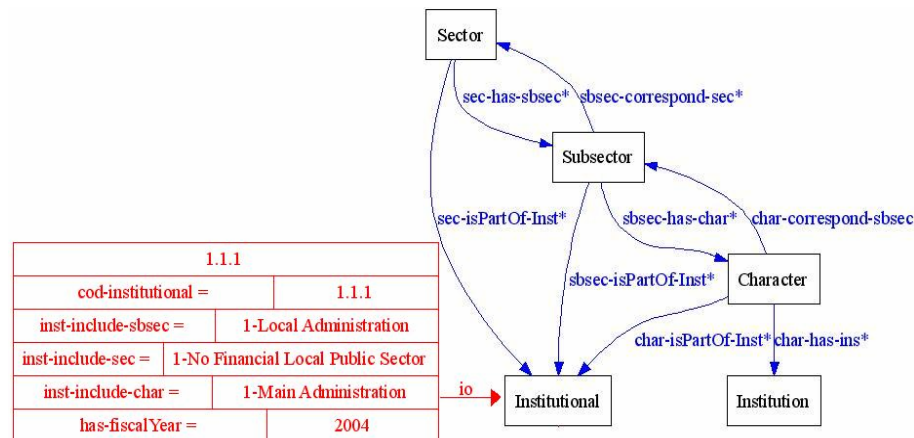


Figure 7: An example of relations between Institutional Classifier Concepts

4.2.9 Implementation: Ontology Verification

To verify the ontology correctness, consistency, completeness and conciseness have to be proved [15].

Consistency. A given definition is consistent if and only if the individual definition is consistent and no contradictory sentences can be inferred using other definitions and axioms. Common errors associated with consistency are: circularity, partition and semantic inconsistency.

Completeness. In fact, it cannot be proved either the completeness of an ontology or the completeness of its definitions, but it can be proved both the incompleteness of an individual definition, and thus deduce the incompleteness of an ontology, and the incompleteness of an ontology if at least one definition is missing regards to the reference framework. So, the ontology is complete if and only if:

- All that is supposed to be in the ontology is explicitly set out in it, or can be inferred.
- Each definition is complete. This is determined by figuring out: (a) what knowledge the definition defines or does not explicitly define about the world; and (b) for all the knowledge that is required but not explicit, check whether it can be inferred using other definitions and axioms. If it can be inferred, the definition is complete. Otherwise, it is incomplete.

Common errors associated with completeness are: incomplete concept classification and partition (subclass partition omission and exhaustive subclass partition omission).

Conciseness. The ontology is concise if it does not store any unnecessary or useless definition, if explicit redundancies do not exist between definitions, and redundancies cannot be inferred using other definitions and axioms. Common errors associated with conciseness are: redundancies of subclass-of-relations, redundancies of instance-of-relations, identical formal definition of some classes and identical formal definition of some instances.

The detection of some of these errors is available in ontology development tools. It was verified its consistency by using Racer [18]. It was very useful to determinate the unsatisfiability problems and their propagation causes. An OWL class is deemed to be unsatisfiable (inconsistent) if, because of its description, it cannot have any instances [37].

During the verification process, experiences of CO-ODE Project [19] [23] and practical experience of teaching OWL-DL reported by [28] have been taken into account. During the ontology development process it is suitable to carry out a permanent and iterative verification process, due to partial verifications allow identifying errors propagation between sets of classes.

4.2.10 Implementation: Validate Competency Questions

Validating the ontology means to check if the ontology fulfill the requirements defined during the specification phase. Then, in order to validate the ontology, it has to be checked if the competency questions are being correctly answered. The evaluation may be performed automatically, if the competency questions are represented formally, or semi-automatically, using specific heuristics or human judgment. To formalize the competency questions RDQL [29] and OWL-QL [12] query languages could be used. RDQL is an implementation of an SQL-like query language for RDF. It treats RDF as data and provides query with triple patterns and constraints over a single RDF model. OWL-QL was designed for query-answering- dialogues among agents using knowledge in OWL. Then, OWL-QL is suitable when it is necessary to carried out an inference in the query.

Since competency questions defined in specification phase did not required complex inferences, we have used the RDQL to validate the ontology. An example of a RDQL query is shown following. This query models the competency question "What are sector and subsector for Main Administration?".

```
SELECT ?x ?y ?z ?nsec ?nsbsec
WHERE (x,<adm:rdsec-hasbsec>,?y)
      (?y,<adm:rdsbsec-has-char>,?z)
      (?z,<rdfn:label>,'1-Administracion Central')
      (?x,<rdfn:label>, ?nsec ),
      (?y,<rdfn:label>, ?nsbsec )
USING rdn FOR http://www.w3.org/2000/01/rdf-schema#
adm FOR http://protege.stanford.edu/
```

In order to evaluate competency questions against the ontology, firstly, a RDF ontology was created from Protégé Project. Then, the queries were implemented using Jena Toolkit [25]. Jena is a Java framework, which provides an API for creating and manipulating RDF models. Jena is open source and grown out of work with the HP Labs Semantic Web Program.

5 An Ontology-based Architecture for Knowledge Sharing

An efficient e-Government service has to allow citizens, businesses and other public departments to have 24 hours access to public information from their home, their offices or even on the move using different access media and devices. This scenario requires that all public authorities are interconnected and that the citizen be able to access public services by a single point even if these services are actually provided by different departments or authorities. To this aim, one requirement is to make available the knowledge stored in public sector. To fulfill this requirement the first step consist on to explicit the knowledge. Using the methodology described previously can carry out this task. The second step is to provide an infrastructure that allows users to access this knowledge. In this paper, we propose to use the architecture shown in Figure 8.

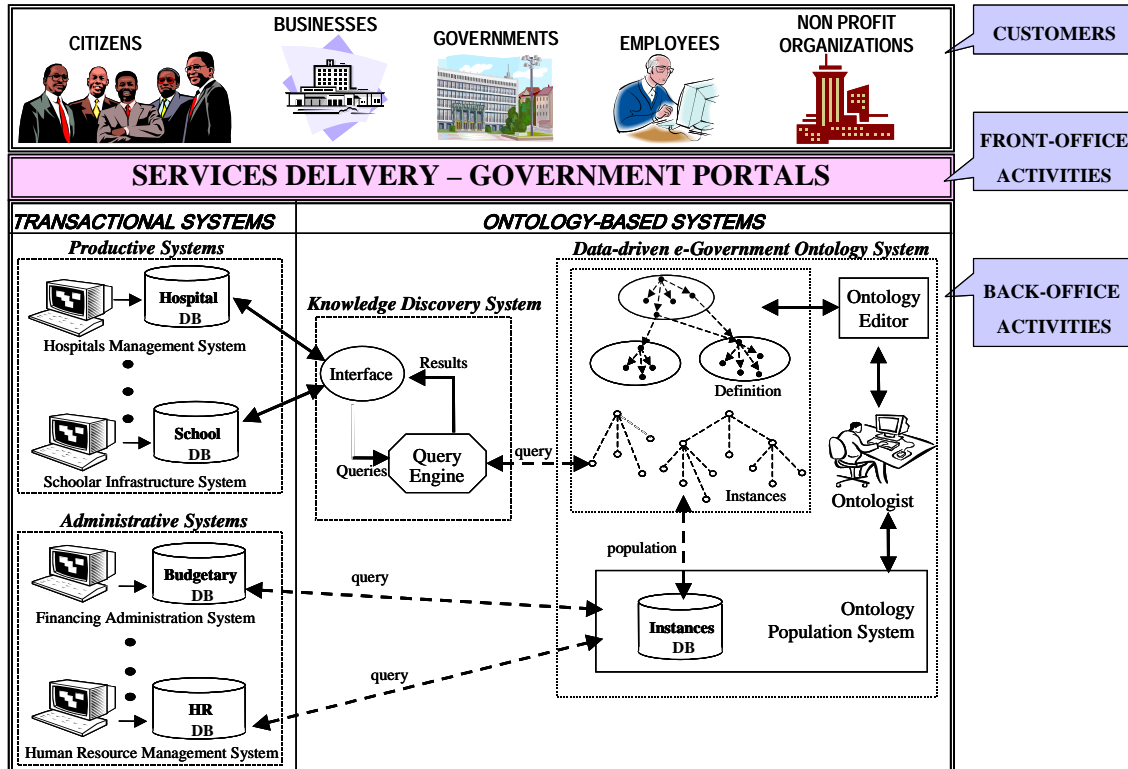


Figure 8: Ontology-based Architecture for Knowledge Sharing in e-Government back-office

An e-Government model could be divided into three main components: Customers, Front-End Activities and Back-Office Activities. These components are described following:

1. Customers: e-government clients are external clients such as citizen, businesses and other governments and, internal clients, specifically public agents who use information and communication technologies to develop their own diary tasks.

2. Front-end Activities: these activities represent the part of a software system that interacts directly with the user. In the government sector an example of a front-end- component is the Government Portal.
3. Back-office Activities: Transactional Systems that could be grouped into Administrative and Productive government systems support these activities. Administrative systems support administrative tasks, while Productive systems manage the information generated by a domain that produces results to external clients (health, education, security). The back-office activities are internal and generally they are unknown for the community. The quality of the services offered by e-government, however, depend on these activities. Then, it is necessary to work on providing an infrastructure to achieve an effective integration and interchange of information managed by back-office activities. To this aim, a knowledge management system could be defined. In this paper, we propose to implement an architecture composed by the following components:
 - e-Government Ontology System: to guide the ontology designer team to carry out the design, implementation, population and maintenance of ontology. An Ontology Editor that supports the design, implementation and maintenance tasks. The ontology population has to be carried out using an Ontology Population System.
 - Knowledge Discovery System: to receive queries from customers and return the results of them through a friendly user interface.With this architecture, a government user or transactional system can access to data and their meanings (semantics), without he/she/it are an authorized user of transactional system.

6 Conclusions and Future Work

In this paper, we have shown how domain experts in public sector can develop their own ontologies merging different methodologies and, software engineering techniques. Particularly, this approach has been used to define General Ontology for the Budgetary Domain in the Santa Fe Province (Argentina), which could be extended by Task Ontologies and used by different government applications.

This work contributes to the semantic interoperability level required in back-office of e-government tasks, bringing about a situation where semantics associated to data is highly specific, complex and used from all government administration areas. The use of ontology-based systems in e-government tasks allow not only to delivery more efficient integrated services but brings the possibility through its front-end, of delivering knowledge included into government areas to community.

As main conclusions that arise by applying the ontology development process, we can mention:

1. A domain in the public sector is characterized by a set of synonyms and terms adopted by users, which are particular to their daily activities that do not have a technical meaning. For that reason, each ontology designer should use the representations that he considers more convenient for the conceptual analysis, performing the ontological agreements when they are required.
2. The application of the process presented in this work for building ontologies has been satisfactory since it has allowed us to verify and document the steps and developed activities, using simple representation tools that are known and practical, have enabling us the communication with the domain experts.
3. Just like in all ICT development projects, it is necessary the support of the highest management levels. But this issue becomes particularly key in ontologies development in the public sector since it is necessary to make ontological agreements among public agents. Also due to the adoption of an ontology-based system implies strong changes in the public agent minds and organizations.
4. The use of general modeling tools coming from Software Engineering constitutes a very important element since Ontology Engineering is improved through tested methods and it is very practical for users at knowledge acquisition, conceptual model modeling and testing stages.

On the other hand, future works are related to:

1. The development of a task ontology for each budgetary life cycle subprocess, considering the domain ontology presented in this paper as a shared vocabulary for the provincial financial administration domain.
2. The development of ontology-based systems for other government domains related to cross-sectional and/or verticals systems, interacting with existing transactional systems as a way to achieve the semantic interoperability in the back-office-, as it proposes in the Figure 8.
3. The formal semantic structure proposed by ontologies could be extended to the e-government portals, allowing the citizens, enterprises and governments to query the ontological systems promoting the knowledge management.

References

- [1] L. Álvarez Sabucedo and L. Anido Rifón, A Proposal for a Semantic-Driven eGovernment Service Architecture, in Proceedings of Electronic Government: 4th International Conference, Copenhagen, Denmark, Lecture Notes on Computer Science Vol. 3591, 2005, pp. 237-248.
- [2] D. Apostolou, L. Stojanovic, T. Pariente Lobo, B. Thoenssen, Towards a Semantically-Driven Software Engineering Environment for eGovernment. in Proceedings of E-Government: Towards Electronic Democracy: International Conference - TCGOV 2005, Computer Science –Vol. 3416, 2005, pp. 157-168.
- [3] W. Arms, D. Hillmann, C. Lagoze, D. Krafft, R. Marisa, J. Saylor, C. Terrizzi and H. Van de Sompel. (2002, January) A Spectrum of Interoperability – The site for Science Prototype for NSDL, D-Lib Magazine, [Online] 8(1). Available: <http://www.dlib.org/>

- [4] D. Brickley and R. V. Guha (2004, February) RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 [Online] Available: <http://www.w3.org/TR/rdf-schema/>
- [5] A. Busson, and A. Keravel, Prospective Study Report on Interoperability in Local eGovernment Projects, Technical Report TERREGOV Project, 2005. Available: www.egovinterop.net
- [6] M. L. Caliusco, A Semantic Definition Support of Electronic Business Documents in e-Collaboration. PhD Dissertation, Universidad Tecnológica Nacional, F.R.S.F, 2005.
- [7] Corcho, O.; Fernández-López, M. and Gómez-Pérez, A. Methodologies, tools and languages for building ontologies. Where is the meeting point? Data & Knowledge Engineering vol. 46, pp 41-64, 2003.
- [8] S. Cranefield, UML and the Semantic Web, in Proceedings First Semantic Web Working Symposium, Stanford University, California, USA, 2001, pp 113-130.
- [9] M. Cristani and R. Cuel, A comprehensive guideline for building a domain ontology from scratch, in Proceedings, International Conference on Knowledge Management (I-KNOW '04), Graz, Austria, 2004.
- [10] R. A. Falbo, Experiences in Using a Method for Building Domain Ontologies, in Proceedings, 6th International Conference on Software Engineering and Knowledge Engineering, International Workshop on Ontology In Action, Banff, Alberta, Canada. 2004, pp. 474-477.
- [11] M. Fernández-López, Overview of Methodologies for Building Ontologies, in Proceedings, IJCAI-99 Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden, 1999.
- [12] R.Fikes, P. Hayes and I. Horrocks, OWL-QL - A Language for Deductive Query Answering on the Semantic Web, Technical Report, KL Laboratory, Stanford University, Stanford, 2003.
- [13] H. Francis, Semantic Interoperability and Citizen/Government Interaction, in Proceedings, Symposium on Geospatial Theory, Processing and Applications, [Online] Ottawa, Canada, 2002. Available: <http://www.isprs.org/commission4/proceedings02/paper.html#2>
- [14] J. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy, and S. W. Tu, The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. International Journal of Human-Computer Studies. Vol. 58, No 1, 2003, pp. 89-123.
- [15] A. Gómez-Pérez, M. Fernández López and O. Corcho, Ontological Engineering with examples from the areas of knowledge management, e-commerce and the semantic web. London: Springer, 2004.
- [16] M. Grüninger and M. S. Fox Methodology for the Design and Evaluation of Ontologies, in Proceedings, IJCAI Workshop on Basic Ontological in Knowledge Sharing, Montreal, Canada, 1995.
- [17] N. Guarino, Understanding, Building and Using Ontologies: A Commentary to Using Explicit Ontologies in KBS Development, International Journal of Human and Computer Studies, 46, 1999, pp 293-310.
- [18] V. Haarslev and R. Möller, RACER System Description, in Proceedings, 1st international Joint Conference on Automated Reasoning IJCAR, Lecture Notes in Computer Science 2083, 2001, pp. 701-706.
- [19] M. Horridge, H. Knublauch, A. Rector, R. Stevens and C. Wroe, A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0, Technical Report, The University Of Manchester Stanford University, 2004.
- [20] IEEE (1996) IEEE Standard for Developing Software Life Cycle Processes. IEEE Computer Society. New York (USA).
- [21] D. Jones, T. Bench-Capon and P. Visser, Methodologies for Ontology Development, in Proceedings, IT&KNOWS Conference, XV IFIP World Computer Congress, Budapest, 1998.
- [22] R. Klischewski and K. Lenk, Understanding and Modelling Flexibility in Administrative Processes, in Proceedings of Electronic Government: First International Conference, Aix-en-Provence, France, 2002, pp 129-136.
- [23] H. Knublauch, M. Horridge, M. Musen, A. Rector, R. Stevens, N. Drummond, P. Lord, N. Noy, J. Seidenberg and H. Wang, The Protégé OWL Experience, in Proceedings, Workshop on OWL: Experiences and Directions, Fourth International Semantic Web Conference, Galway, Ireland, 2005.
- [24] O. Lassila and D. McGuinness, The Role of Frame-Based Representation on the Semantic Web, Knowledge Systems Laboratory, Stanford University, Technical Report KSL-01-02, 2001.
- [25] B. McBride, Jena: A Semantic Web Toolkit. IEEE Internet Computing. November/December 2002 Vol. 6, No. 6, 2002, pp. 55-59.
- [26] P. Miller (June 2000) Interoperability - What is it and Why should I want it?, Ariadne Issue 24. [Online] Available: <http://www.ariadne.ac.uk/>
- [27] N. Noy, D. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, Knowledge Systems Laboratory, Stanford University, Technical Report KSL-01-05, 2001.
- [28] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, C. Wroe (2004) OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns, in Proceedings, Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, Lecture Notes in Computer Science 3257, Whittlebury Hall, UK, 2004, pp 63-81
- [29] A. Seaborne (2004, January) RDQL - A Query Language for RDF, W3C Member Submission 9. [Online] Available: www.w3c.org
- [30] A. P. Sheth, Changing Focus on interoperability in information systems: from system, syntax, structure to semantics. In Interoperating Geographic Information Systems, (Goodchild, M., Egenhofer, M., Fegeas, R. & Kottman Eds, C.), Kluwer Academic Publishers, 1999, pp. 5-30.
- [31] M. Smith, C. Welty and D. McGuinness (2004, February) OWL Web Ontology Language Guide, W3C Recommendation. Available: www.w3c.org
- [32] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke. OntoEdit: Collaborative ontology development for the semantic web. University of Karlsruhe, Institute AIFB, Technical Report 419, 2002.

- [33] R. Traunmüller and M. Wimmer, KM for Public Administration: Focusing on KMS Feature Requirement, in Proceedings, Practical Aspects of Knowledge Management, 4th International Conference. Springer Verlag. Berlin, LNAI 2569, 2002, pp 314-325.
- [34] Unified Modeling Language (2007, March). Available: <http://www.uml.org/>
- [35] M. Uschold, Building Ontologies: Towards a Unified Methodology, in Proceedings, Expert Systems the 16th Annual Conference of the British Computer Society Specialists Group on Expert Systems, Cambridge, UK, 1996, pp 16-18.
- [36] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner, Ontology-Based Integration of Information –A Survey of Existing Approaches, in Proceedings IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, pp. 108-117.
- [37] H. Wang, M. Horridge, A. Rector, N. Drummond and J. Seidenberg, Debugging OWL Ontologies: A Heuristic Approach, in Proceedings 4th International Semantic Web Conference (ISWC'05), Galway, Ireland, (2005).
- [38] M. Wimmer, A European perspective towards online one-stop government: the eGov project, Electronic Commerce Research and Applications, Vol.1(1), 2002, pp. 92-103.