A Semantic Cooperation and Interoperability Platform for the European Chambers of Commerce

Michele Missikoff and Francesco Taglino

1 Introduction

The LD-CAST project aims at developing a semantic cooperation and interoperability platform for the European Chambers of Commerce. Some of the key issues that this platform addresses are:

- The *variety* and *number* of different kinds of resources (i.e., business processes, concrete services) that concur to achieve a business service
- The *diversity* of cultural and procedural models emerging when composing articulated cross-country services
- The limited possibility of reusing similar services in different contexts (for instance, supporting the same service between different countries: an Italian–Romanian cooperation is different from an Italian–Polish one)

The objective of the LD-CAST platform, and in particular of the semantic services provided therein, is to address the above problems with flexible solutions. We aim at introducing high levels of flexibility, both at the time of development of business processes and concrete services (i.e., operational services offered by service providers), with the possibility of dynamically binding c-services to the selected BP, according to user needs. To this end, an approach based on semantic services and a reference ontology has been proposed.

Recently, there has been an acceleration in the use of ontology-based solutions aimed at supporting the cooperation and interoperability among different organizations. Diversities typically lie in the different levels of automation and, when automated, in the characteristics of the information systems having diverging architectures and data format. An ontology-based approach can help in solving the above problems.

M. Missikoff (✉)
National Research Council, Institute of Systems Analysis and Informatics "Antonio Ruberti", Viale Manzoni 30, 00185 Rome, Italy

T. Vitvar et al. (eds.), *Semantic Technologies for E-Government*,
DOI 10.1007/978-3-642-03507-4_6,

The role of a reference ontology is a central one. In fact, an ontology is "*a formal, explicit specification of a shared conceptualisation*" (Gruber 1993); therefore, it represents:

- The result of a consensus process reached among a group of domain experts (shared conceptualisation), possibly obtained by using focused methods, like a folksonomy, and involving a wider community
- A formal specification, and as such it can be interpreted by a machine

According to that, an ontology is usually adopted:

- At a social level, as a common reference for improving the communication and interoperability among (human or software) members of the same community or of different communities
- At a computational level, for supporting automatic reasoning and *intelligent* querying activities

In the proposed approach we have different components, defined at different levels of abstraction, to achieve a coherent scenario. In particular, we make use of the following notions (see also Fig. 1):

Business Episode. This corresponds to a *user request* (e.g., request a business certificate) that typically falls into a predefined category.

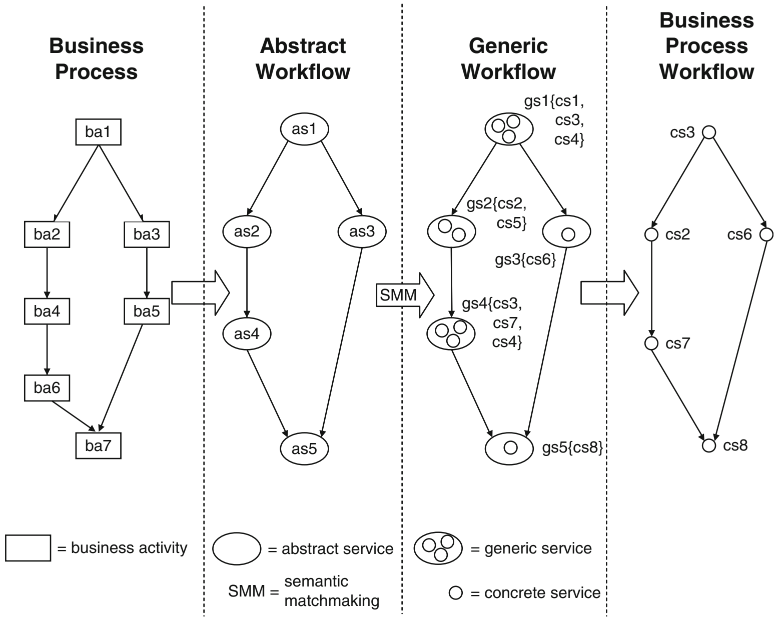

Fig. 1 The transition from business process to business process workflow

Business Process (BP). This is an abstract, generic specification (not yet executable), corresponding to a *Business Episode*. A repository of generic BPs represents the starting point for the flexible composition of the customized workflows.

Abstract Workflow (AW). It is a first level of implementation of a BP. The AW is represented through an executable language, but it is not yet fully executable, since concrete services are not yet referenced here. In general, for each BP there can be more than one abstract workflow (but here we will consider a one-to-one correspondence).

Business Activity (BA). The component of a Business Process.

Abstract Service (AS). The component of an Abstract Workflow. The correspondence between business activities and abstract services is not one to one. For instance, in Fig. 1, the business activities *ba3* and *ba5* are replaced by the only abstract service *as3*.

Semantic Annotation. It is the method adopted to define the semantic specification of an activity. A semantic annotation is built as a vector of concepts selected from the reference ontology. Therefore we refer to it as an ontology-based feature vector[1] (OFV).

Concrete Service (CS). It is an executable service registered by a service provider in the concrete service registry (CSR). A concrete service (CS) can be automatic (e.g., a web service) or manual (e.g., desk services). The stored information is about the name, a natural language description of the service functionality, the service provider, the execution time and price, and where to access the service. This last information is given by an URL in the case of an automatic service, and by the physical address of the office providing the service in the case of a manual service.

Generic Service (GS). The GS of a business activity, is the set of concrete services that, with respect to the semantic similarity proposed in LD-CAST, are equivalent in the way they execute the a given abstract service.

Generic Workflow (GW). It is obtained starting from the abstract workflow and replacing all the abstract services with generic services.

Business Process Workflow (BPW). It is an executable version of the abstract workflow, where a single concrete service of a generic service is bound to the relative abstract service.

The objective of the LD-CAST platform is to allow a flexible and automatically supported transition from the business process to the business process workflow (Fig. 1). In particular, the transition from the abstract workflow to the generic workflow is guided by a semantic matchmaking mechanism.

[1] In this application, the components of the OFV are not intended to be ordered. However, ordering the components of the OFV will allow a first level of relevance of the features, with respect to a single OFV to be expressed.

2 The LD-CAST Phased Semantic Approach

The LD-CAST approach is based on 4 main phases: two *setup time* phases and two *run-time* phases, as summarized in the Table 1 and briefly described afterwards.

2.1 Phase 1: Application Resources Set Up

During this first phase the setting up of knowledge resources is performed. The two major activities of this phase are:

(a) *Business processes design (business level)*. BPs are designed by the Chambers of Commerce, with the support of knowledge engineers, by using the ADOe-Gov modeling tool and language (LD-CAST 2006). Corresponding abstract workflows, which are executable code with no attached executable services (concrete services), are also defined. However, for each concrete service to be called, a labeled placeholder is put in the AW as an abstract service.

(b) *Registration of concrete services*. Service providers register concrete services (web services or desk services) in the LD-CAST platform, by filling in pre-defined templates. In particular, information about where each service can be accessed are registered (through the WSDL in the case of a web service and through the physical address in the case of a desk, i.e., manual, service).

2.2 Phase 2: Development of Semantic Contents

During this phase the development of the semantic content repository is performed. The two major activities of this phase are the following (Fig. 2):

(a) *Ontology Creation*. The LD-CAST reference ontology is about the services made available through the involved CoC. It has been built by the Chambers of

Table 1 The four phases of the LD-CAST approach

	Phases	Activities
Setup	(1) Application resources set up	• Business processes modeling • Abstract workflow definition • Identification and registration of concrete services
	(2) Development of semantic contents	• Ontology creation • Semantic annotation of resources
Runtime	(3) Business process work-flow set up	• User's request submission • Business process workflow completion
	(4) Business process work-flow (BPW) execution	• Actual BPW execution • Monitoring of the actual execution

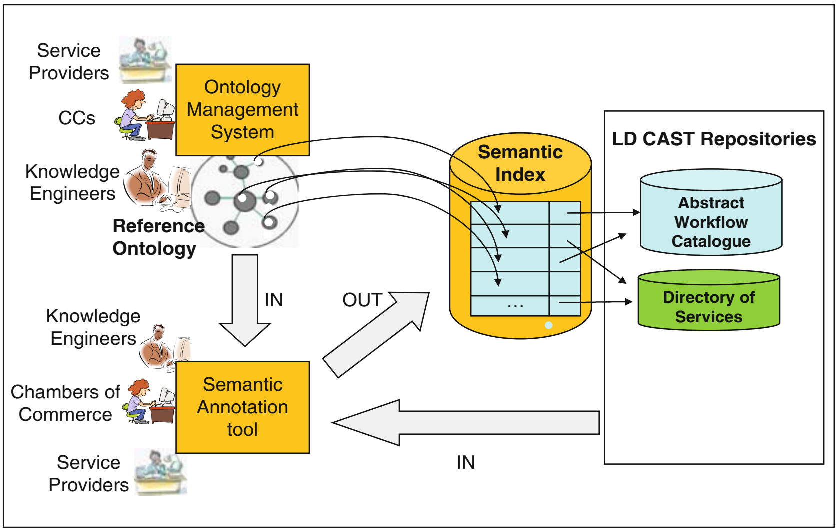

Fig. 2 Development of semantic contents

Commerce, with the support of knowledge engineers, by using the Athos Ontology Management System (ATHENA 2005), based on the OPAL (Object, Process Actor Language) ontology representation method (Missikoff et al. 2007). The ontology is built using a step-wise approach and, in particular, adopting the UPON (Unified Process for ONtology building) methodology (De Nicola et al. 2009).

(b) *Semantic annotation of knowledge resources*. The purpose of the semantic annotation is to associate an ontology-based semantic expression (in our approach an OFV) to an operational resource (i.e., abstract service or concrete service). The objective of this task is to describe all the stored resources in terms of the reference ontology, in order to establish semantic relationships among them.

Summarizing, at the end of the Design Phases we will have the following resources available:

- BP developed by the CoC and abstract workflows
- Registry of the available concrete services to be used to actually execute the AWs
- Ontology of the CoC services
- Semantic indexes consisting in the semantic annotation of concrete services and abstract services (tasks)

Such digital resources will be used in the run-time phases as explained below.

2.3 Phase 3: Business Process Workflow Setup

This phase is triggered by the submission of a request by the end user. The end user of the LD-CAST platform is someone (e.g., an entrepreneur) who refers to a CoC to achieve a business service. A user request is matched by a BP. Starting from the selected BP, the platform identifies the corresponding AW and searches for the required concrete services for transforming the AW into an executable BPW (see Fig. 3). This phase is articulated into the following steps.

(a) *User's request submission and business episode selection*. A user submits a request asking for a business service corresponding to one of the registered business processes.

(b) *Semantic search and discovery for business process workflow completion*. For each BP a corresponding abstract workflow has been previously implemented. In essence, an abstract workflow represents the control flow, with the sequencing of operations but without concrete implementations of the required activities (i.e., concrete services), therefore a business process workflow (BPW) completion is necessary. We refer to BPW completion as the dynamic binding of a business process to concrete services that makes the former executable by a workflow engine (see Fig. 4). This BPW completion is supported by a semantic similarity matchmaking mechanism that, reasoning on the ontology and using the semantic annotation performed on concrete services and abstract services, identifies the most suitable concrete services (automatic or manual) able to perform the abstract services of an AW. In particular, the application of the semantic matchmaking determines the identification of the generic workflow, where each abstract service is replaced by a generic service, which is a set of concrete service. All the concrete services of a generic service are intended to be semantically equivalent and then suitable to be bound to the corresponding abstract service. The further selection of a single concrete service from each generic service, by using non functional criteria (like for instance execution

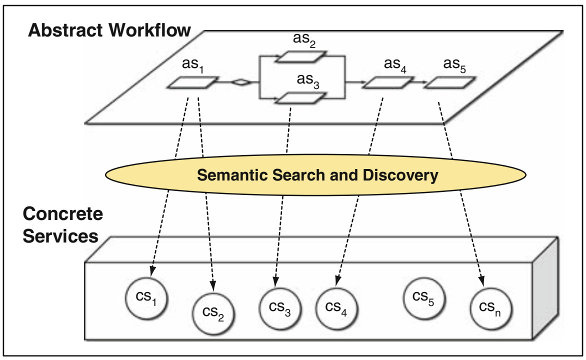

Fig. 3 Binding of concrete services

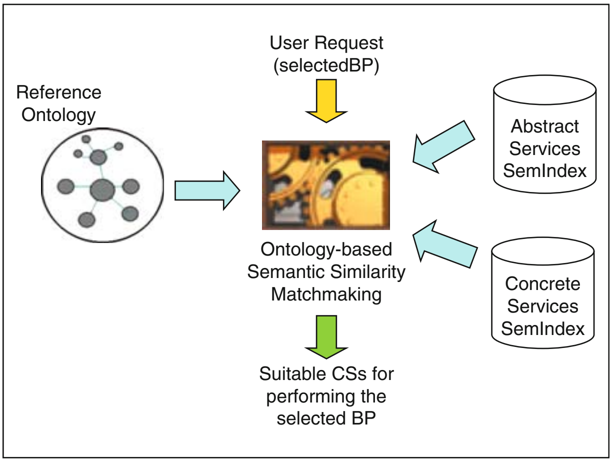

Fig. 4 Semantic search and discovery of concrete services

time and price of the services) will allow the final generation of the BPW that is fully executable.

(c) *Business process workflow release to the LD-CAST workflow engine*. After the binding of concrete services to the abstract services, the complete business process workflow is ready for being executed by the LD-CAST process execution engine.

2.4 *Phase 4: Business Process Workflow Execution*

This phase concerns the actual execution and monitoring of the business process workflow. The activities indicated in the AW will be implemented by the concrete services selected in the previous phase.

An abstract workflow, and the corresponding final BPW, is represented by using an executable language to this end we adopted BPEL4WS.[2] Then, the BPEL code is fed to a BPEL engine (such as Active BPEL[3]) for the actual execution, including the invocation of the concrete services. Please note that this approach accepts also the binding to manual (desk) services. To this end, we represent the latter with proxy web services (a sort of "empty" web service) that prompt the users for the actual execution of the requested manual operations.

[2] http://www.ibm.com/developerworks/library/specification/ws-bpel/

[3] http://sourceforge.net/projects/activebpel

2.5 Summarizing the Role of the LD-CAST Ontology

In Fig. 5 a representation of the semantic streamline of the LD-CAST platform is outlined. The figure is mainly to highlight the position and the role of the semantics-based components into the LD-CAST approach. For this purpose, the semantics-based components are bold rounded, to be easily distinguished, while the other components of the LD-CAST platform are not fully represented. The picture focuses on the semantics-based components, their interactions and how they concur to the LD-CAST platform and solution. According to the previous section, the picture is divided into Design Time and Run Time, and the different software components are positioned accordingly.

- *Athos* is the ontology management system which provides support for the ontology construction. Athos is based on the OPAL (Missikoff et al. 2007) ontological framework for the knowledge representation. We wish to recall that the LD-Cast ontology has been developed by applying the ontology engineering method UPON (De Nicola et al. 2009).
- *Semantic Annotation Environment* provides the support for semantically annotate the Business Processes (more precisely, the activities composing them) and the concrete services that are registered and made available to the LD-CAST platform. Such components acquire the ontology from Athos, and the abstract services and concrete services from their respective repositories. Finally, the defined semantic annotation expressions are stored as semantic

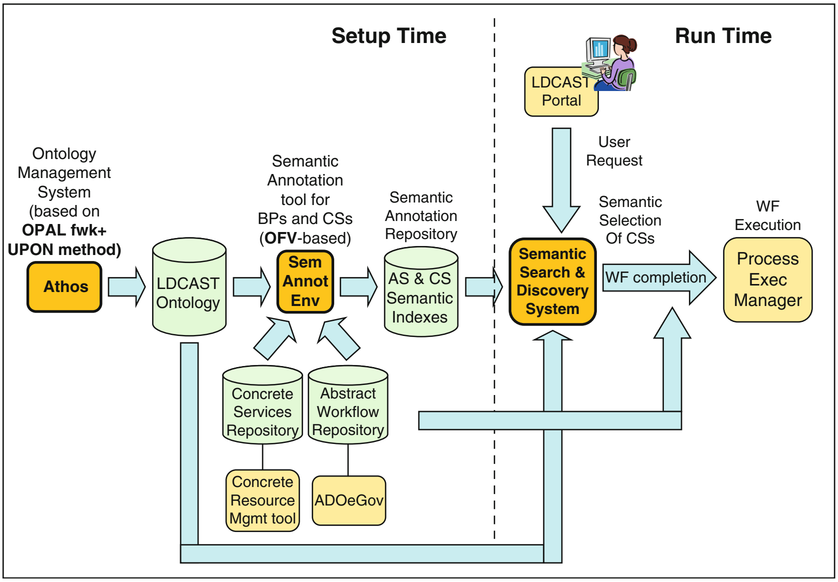

Fig. 5 The LD-CAST semantic streamline

indexes. Semantic annotation expressions in LD-CAST are represented by Ontology-based Feature Vectors (OFV) (Missikoff et al. 2008). An OFV consists of a set of concepts from the ontology that characterize a given resource, which can be a concrete services or an abstract service. From the semantic point of view, the setup phase produces the following two main artifacts: the reference ontology and the semantic annotation of abstract services and concrete services.

- *Semantic Search and Discovery System (SS&DS)* provides a semantics-based support to search and retrieval of concrete services suitable for making the selected business process executable. The selection of the business process (and consequently of the corresponding abstract workflow) triggers the SS&DS. Then the SS&DS searches for the concrete services suitable to be bound to the abstract services composing the abstract workflow. This search is performed through a semantic similarity matchmaking mechanism, which accesses and uses the semantic resources previously built (the reference ontology and the semantic annotations).

As briefly recapped above, the ontology is a key component of the LD-CAST approach and it plays an important role both at the setup and runtime. At setup time, it is used for the semantic annotation of the abstract services composing the abstract workflow and the concrete services. This is a very crucial step since it allows semantic correspondences between the abstract services, which identify what has to be done (i.e., *submitting the request for a fiscal verification*), and the concrete services, which represent actual available implementations be established. BPs, which are the specifications for the abstract workflows, are modeled by the Chambers of Commerce, while the concrete services can be made available by any kind of service provider. Consequently, their descriptions are not homogeneous and, in principle, they are not aligned with the definition of the activities specified in the BP, and more in particular with the abstract services that compose the abstract workflows. The semantic annotation addresses this issue by using the ontology as a semantic bridge capable of filling (large part of) the divergences between these two heterogeneous artifacts (ASs and CSs).

The objective of the SS&D service is to identify the compatibility of two resources, say an abstract service and a concrete service, by analyzing their respective semantic annotations. The analysis is accomplished by computing a semantic similarity between the OFV, and it returns the measure of how much a given concrete services is suitable to execute a given abstract service. Such a semantic coupling is executed at run-time, in the Phase 3, as described in the previous section.

At runtime, during the search and retrieval of concrete services to be bound to the abstract services (Workflow completion), the role of the ontology is crucial as well, and it is used at two different levels:

- *Explicitly*: when the ontology is directly accessed by the Semantic Search and Discovery (SS&D) component for querying its semantic content (e.g., Specialization/Generalization hierarchies, concept weight).

- *Implicitly*: when the ontology-based semantic annotation expressions, built in terms of the reference ontology, are used by the Semantic Search and Discovery subsystem, for coupling abstract services and concrete services.

In LD-CAST, the original method for semantic annotation has been developed, referred to as Ontology-based Feature Vectors. This method will be presented in details in Sect. 3.

3 OFV-Based Annotation in LD-CAST

Semantic annotation is a technique used to describe the meaning of a digital resource by using a reference ontology. This technique is attracting attention for its great potentiality in the Semantic Web (Berners-Lee et al. 2001).

Among typical applications of semantic annotation, we can find:

- *Semantic retrieval*, i.e., the possibility of retrieving digital resources (not only documents) on the basis of their semantic content. This service is at the basis of the following, more specific applications
- *Document management*, for the semantics-based organization and retrieval of digital documents
- *Knowledge management*, for organization and retrieval of enterprise knowledge
- *Web services publishing and discovery*, with semantic matchmaking of requested and offered services
- *Semantic interoperability*, by annotating local resources (information and processes) to support business cooperation among enterprise software applications

In the LD-CAST approach, semantic annotation is at the basis of the semantic search and retrieval of digital resources. In particular, the semantic search is here used to select concrete services suitable for making abstract workflows fully executable.

When we analyzed the possible alternatives in representing a semantic annotation, we considered the following issues:

- *Kind of annotated resources*: i.e., unstructured documents (e.g., textual documents), structured documents (e.g., data schemas), any type of model, web services. In LD-CAST, the annotation process focuses on the annotation of abstract services, and service implementations (human or automatic).
- *Type of user*: human oriented or machine oriented annotation; in LD-CAST, the annotation is mainly for machine oriented consumption, since it is used to support an automatic matchmaking algorithm.
- *Level of formality*: in general, annotations may vary from informal, by using (controlled) natural language, to structured, by using tables and diagrams, to formal, by using formal language-based expressions. In LD-CAST, the annotation is structured as it is represented as Ontology-based Feature Vectors.
- *Terminology restriction*: here there are at least three alternatives: absence (no restriction); mandatory (fully controlled terminology); advised (suggested, but

not mandatory). In the LD-CAST approach, the terminology restriction is mandatory, since the annotation is fully restricted by the terms used in the ontology.
- *Positioning*: embedded to the annotated resource (it needs to modify the original resource); attached (stored in a separated repository). In LD-CAST, an attached approach, through the definition of semantic indexes, has been followed since it is less intrusive with respect to the original resource.

The semantic annotation method used in LD-CAST is based on the usage of OFV. Essentially, an OFV is composed by concepts from the ontology, each concurring to characterize the annotated digital resource. More in general, in such an approach, features vectors are also used to represent requests from the user searching for digital resources. Even if the structure is the same, to avoid confusion, we will refer to it as a *request feature vector* (RFV). A RFV is very similar to the popular way used when searching with Google, where a list of terms is defined. The main difference here is that the RFV cannot use any set of terms, but it is restricted to the labels of the concepts in the ontology.

In LD-CAST, we annotate both concrete services and abstract services, for automatically supporting the workflow completion.

When coupling concrete services to abstract services, three main scenarios have been identified (see Fig. 6). Each scenario is characterized by a different degree of usage of the ontology, which depends on the existing relationships and dependencies among the ontology, the abstract workflows (but also the business processes) and the concrete services:

- *Trivial*: concrete services and abstract services are strictly related; there is a one-to-one matching between CSs and ASs. In this scenario, there is no need of semantic annotation or probably the semantic annotation is trivial: $|ofv_{as}|=1$ and $|ofv_{cs}|=1$, where ofv_{ba} and ofv_{cs} are the OFV for *as* and *cs*, respectively an abstract service and a concrete service.
- *Simple*: there is a one-to-one correspondence between *as* and a concept in the ontology, since the construction of the ontology has been influenced by the BPs definition. Conversely, the CSs have been defined independently. In this case we have $|ofv_{as}|=1$ and $|ofv_{cs}|=n$. Essentially, it means that, in general, there is not only one concept catching the entire semantics of a given concrete service.
- *Full*: ASs and CSs are independently built and both of them are independent with respect to the ontology. In this case we have $|ofv_{as}|=n$ and $|ofv_{cs}|=m$.

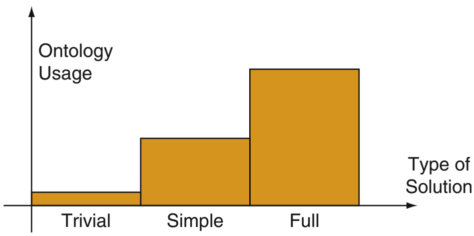

Fig. 6 Ontology usage levels

In the LD-CAST project, we decided to implement the *Simple* case, since it mainly reflects the scenario of the project itself where the construction of the ontology is strongly influenced by the modeled BPs.

Another important feature of the proposed method is represented by the use of a *Weighted Reference Ontology* (WRO), containing the relevant concepts in the given domain. Here, we wish to recall the definition of an ontology, taken from OMG Ontology Definition Metamodel (DSTC 2005):

> An ontology defines the common terms and concepts (meaning) used to describe and represent an area of knowledge. An ontology can range in expressivity from a Taxonomy (knowledge with minimal hierarchy or a parent/child structure), to a Thesaurus (words and synonyms), to a Conceptual Model (with more complex knowledge), to a Logical Theory (with very rich, complex, consistent and meaningful knowledge).

We restrict our view of the ontology to a taxonomy of concepts. This simplified view in our proposal is enriched with a weight, associated with each concept, representing its *featuring power*, which indicates how much such a concept is selective in characterizing the available digital resources. In accordance with the Information Theory (Shannon 1948) a concept weight will be used to determine the (relative) information content of the concept itself. A high weight corresponds to a low selectivity level, meaning that many resources are characterized by the concept. Conversely, a low weight corresponds to a high selectivity, and therefore its use in a request will be more significant.

4 The Weighted Reference Ontology

As anticipated in the previous section, the proposed method is based on a Weighted Reference Ontology, where each concept is enriched with a weight.

The way of associating such weights derives from the structure of the specialization hierarchy and follows a top down approach: the weights are assigned starting from the root of the hierarchy.

In order to have a unique top concept in the specialization hierarchy, *Thing* is added as the root of the hierarchy. *Thing* represents the most general concept in the ontology. Whatever is the specialization hierarchy, *w(Thing)* will always be equal to 1.

As weighting strategy, we adopted a uniform distribution.[4] Therefore, given a concept c_x, having weight equal to $w(c_x)$, the weight of each of the child c of c_x, $w(c)$, is

$$w(c) = w(c_x)/|children(c_x)|$$

[4]We are currently analyzing possible other ways of assigning weights to the concepts of the ontology. For instance, considering the relative frequency of each concept with respect to the built OFVs.

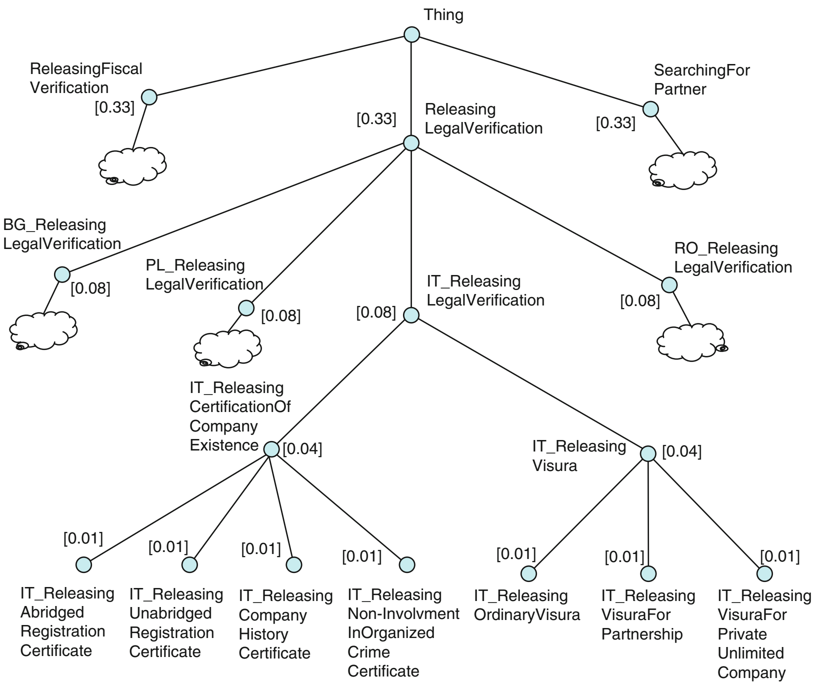

Fig. 7 A fragment of the LD-CAST weighted reference ontology

For instance, consider Fig. 7, assumed that *w(Thing)* = 1, therefore

$$w(ReleasingFiscalVerification) = 1/3,$$

$$w(ReleasingLegalalVerification) = 1/3, \textit{ and}$$

$$w(SearchingForPartners) = 1/3.$$

The rationale behind the presented method for the assignment of the weights is that the weight of a concept expresses how frequently a concept appears in (characterizing) the available digital resources or in other words, the probability of the concept to annotate a resource. For this reason, the leaves in the taxonomy, which represent very specific concepts, are assumed to have the lowest weight (and therefore the highest selectivity factor). For instance, with respect to Fig. 7, there will be a higher probability to annotate a concrete service *cs* with the generic concept *ReleasingLegalVerification*, than to annotate *cs* with *ReleasingOfNon-InvolvmentInOrganizedCrimeCertificate* that is a more specific activity.

In particular, Fig. 7 shows an excerpt of the LD-CAST WRO. This WRO is about three kinds of business services, Legal Verification, Fiscal Verification and

Search for Partners. Furthermore, the WRO describes these three business services for the four countries involved in the project i.e., Bulgaria, Italy, Poland and Romania. The excerpt mainly shows the part of the WRO concerning the Italian Legal Verification.

5 Semantic Search and Discovery in LD-CAST

The role of the Semantic Search and Discovery in LD-CAST is to identify, through a semantic matchmaking mechanism, the concrete services that are suitable for performing the abstract services composing an abstract workflow. This will allow the completion of the abstract workflow into an executable representation via the binding of concrete services.

Since there can be more than one concrete service *cs* suitable for a given abstract service *as*, the semantic search will return, for each *as*, a set of CSs represent a generic service.

Essentially, a Generic Service (GS) is a class of equivalence of concrete services with respect to the semantics of the corresponding *as*.

The fact that for each abstract service there will be a corresponding GS means that, given an abstract workflow *aw*, more than one combination of CSs can be suitable for performing *aw*. The selection of only one out of them is out of the scope of the semantic search. In fact, from a semantic point of view, all the CSs of a GS are considered equivalent.

5.1 The Semantic Matchmaking Mechanism

The Semantic Matchmaking mechanism is based on the identification of a similarity degree between an AS and each CS in order to identify the corresponding GS. All the CSs having a similarity degree equal or higher to a fixed threshold will be in the GS of the considered AS. The semantic matchmaking works on the semantic annotation previously associated to the ASs and the CSs.

The matchmaking mechanism proposed in this approach is based on a semantic similarity technique (Maguitman et al. 2005). A brief comment on the added value of a search mechanism based on semantic techniques and a similarity measure is here reported.

5.1.1 "Semantic Search" Vs. "Keyword-Based Search"

Commonly used search engines are based on string matching techniques. Even if they can be highly sophisticated (i.e., Google) they are only able to compare keywords and finding a match among them. Semantic search techniques, supported by the use of a reference ontology, aim to go beyond keyword-based search. In fact, an ontology defines concepts that are defined not only by labels, but especially by semantic relationships (i.e., specialization/generalization) which can be consequently exploited during the search activities. For instance, considering the fragment of the LD-CAST ontology reported in Fig. 7, we have that the *IT_ReleasingVisura* and the *IT_ReleasingCertificateOfCompanyExistence* concepts are two specializations of the *IT_ReleasingDocumentofCompanyExistence* concept. This means that whenever we are searching for a document about the existence of a given Italian company, we can use a service previously annotated with either a *IT_ReleasingVisura* or a *IT_ReleasingCertificateOfCompanyExistence* since both are specialization of *IT_ReleasingDocumentofCompanyExistence* process.

Please note that, in closed environments, even if very wide, like the LD-CAST community is, such an approach is much more usable than in very open and very heterogeneous environments like Internet is.

5.1.2 "Similarity Matching" Vs. "Exact Matching"

Exclusive exact matching algorithms allow to retrieve only resources that fully match the user request. Nevertheless, partial results can be better than no results and give greater flexibility to the user. In the LD-CAST proposed method, a similarity matchmaking has been introduced which is able to evaluate a similarity measure of available services with respect to the request. The tolerance of the method is defined by a threshold fixed by the user. If for a given resource, the similarity is above a fixed threshold, the resources are retrieved as a satisfying result.

5.2 The Semantic Matchmaking Algorithm

Given a WRO, the notion of similarity here adopted relies on the probabilistic approach defined by Lin (Lin et al. 1998), which is based on the notion of information content and entropy. According to the standard argumentation of information theory, the information content of a concept c_x is defined as:

$$- \ln w(c_x)$$

that is, as the weight of a concept increases the informativeness decreases, therefore the more abstract a concept the lower its information content is. In fact, the weights decrease from the top to the bottom of the specialization hierarchy and whatever is c_x, $0 \leq w(c_x) \leq 1$, we have that $\ln(w(c_x))$, the natural logarithm of $w(c_x)$, has an inverse behavior in the sense that it decreases from the bottom to the top. Essentially, since a leaf is a very specific concept, annotating a resource with a leaf, is considered more relevant than annotating a resource with a more general concept. Finally, $\ln(w(Thing)) = 0$ because *Thing* is the top of the specialization hierarchy and its informative content is null.

Given two concepts c_x and c_y, their concept similarity *consim*(c_x,c_y), is defined as the maximum information content shared by the concepts divided by the information content of the two concepts (Jaccard 1901). Formally, since we assumed that the taxonomy is a tree, the *least upper bound* (*lub*) of c_x and c_y, *lub*(c_x,c_y), is always unique and defined and provides the maximum information content shared by the concepts. Therefore, *consim* is the similarity function between concepts, based on the Lin (Lin et al. 1998) similarity. Let be c_x and c_y two concepts[5]

$$consim(c_x,c_y).= 2*ln(w(lub(c_x,c_y)))/(ln(w(c_x))+ln(w(c_y)))$$

Then, whatever are c_x and c_y, $0 \leq consim(c_x,c_y) \leq 1$.

The semantic matchmaking algorithm assumes that the semantic annotations of ASs and CSs have been already performed and in particular, according to the *Simple* scenario:

- Each abstract service *as* is annotated with only one concept ($/ofv_{as}/=1$);
- Each concrete service *cs* can be annotated with more than one concept from the ontology ($/ofv_{cs}/=n$).

The Semantic Matchmaking algorithm assumes that:

- **w(c_x)** is the weight associated to the concept c_x;
- **ln(w(c_x))** is the natural logarithm of the weight of c_x;
- **lub(c_x,c_y)** is the least upper bound between c_x and c_y with respect to the specialization/generalization hierarchy, that is the nearest common ancestor of both c_x and c_y. For instance, with respect to Fig. 1 the least upper bound of *IT_ReleasingAbridgedRegistrationCertificate* and *IT_ReleasingOrdinaryVisura* is *IT_ReleasingLegalVerification;*
- ***semsim*** is the similarity function that calculates the similarity degree between an AS and a CS. The *semsim* function refers to the semantic annotation of the ASs and the CSs.

Let as_n be an abstract service with $ofv_{asn}=\{c_j\}$
Let cs_m and ofv_{csm} be respectively a concrete service and its OFV, with $/ofv_{csm}/= k$

$$semsim(as_n,cs_m) = semsim(ofv_{asn},ofv_{csm}).$$

Whatever ba_n and cs_m are, $0 \leq semsim(ba_n, cs_m) \leq 1$.
In particular, since with respect to the three scenarios of Sect. 3 we claimed to address the *Simple* one, ofv_{asn} is composed by only one concept c_j, the *semsim* function is calculated by evaluating the average value of the similarity degree between c_j and each concept in the ofv_{csm}

$$semsim(ofv_{asn},ofv_{csm}) = (\sum consim(c_j, ofv_{csm[i]}))/k \text{ with } i = 1\ldots k$$

Now, assuming a business process with two business activities about the verification of the legal and fiscal status of an Italian company (Fig. 8), and the

[5]Note that the algebraic sign can be omitted in such rational expression

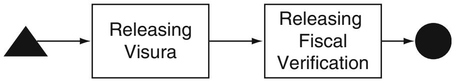

Fig. 8 An example of business process

Table 2 Weight of concepts in the ontology

c	w(c)
Thing	1
ReleasingFiscalVerification	0.33
ReleasingLegalVerification	0.33
SearchingForPartner	0.33
BG_ReleasingLegalVerification	0.08
PL_ReleasingLegalVerification	0.08
RO_ReleasingLegalVerification	0.08
IT_ReleasingLegalVerification	0.08
IT_ReleasingCertificateOfCompanyExistence	0.04
IT_ReleasingVisura	0.04
IT_ReleasingAbridgedRegistrationCertificate	0.01
IT_ReleasingUnabridgedRegistrationCertificate	0.01
IT_ReleasingCompanyHistoryCertificate	0.01
IT_ReleasingOrdinaryVisura	0.02
IT_ReleasingVisuraForPartnership	0.02

corresponding AW with two abstract services with the same labels, and the fragment of the ontology in Fig. 7, we annotate the abstract service *ReleasingVisura* with the concept *IT_ReleasingVisura*. According to that, the OFV annotating the abstract service is:

$$ofv_{RLV} = [IT\_ReleasingVisura]$$

On the other hand, we here provide the annotation of 4 concrete services, *cs1*, *cs2*, *cs3* and *cs4*.

$$\begin{aligned}
ofv_{cs1} &= [IT_ReleasingOrdinaryVisura,\\
&\quad IT_ReleasingCompanyHistoryCertificate]\\
ofv_{cs2} &= [IT_ReleasingLegalVerification,\\
&\quad BG_ReleasingLegalVerification]\\
ofv_{cs3} &= [IT_ReleasingOrdinaryVisura,\\
&\quad IT_ReleasingVisuraForPartnership]\\
ofv_{cs4} &= [IT_ReleasingAbridgedRegistrationCertificate]
\end{aligned}$$

According to Fig. 7 we have the following weight for each concept *c* (Table 2).

Now, the *consim* function between *IT_ReleasingLegalVerification*, which is the only concept in ofv_{RLV}, the OFV annotating *the ReleasingLegalVerification* abstract service, and any concept *c* belonging to at least one OFV annotating a concrete service, is computed (Table 3).

Finally, the *semsim* function between ofv_{RLV} and the OFV relative to the CSs is computed (Table 4).

Furthermore, assuming that the threshold has been fixed at 0.65 (shown by the horizontal dashed line in Table 4), we have that the resulting GS for the abstract

Table 3 Consim values between c_3 and all the other concept c in an ofv

c	Consim (*IT_ReleasingVisura*, c)
IT_ReleasingOrdinaryVisura	0.90
IT_ReleasingCompanyHistoryCertificate	0.64
IT_ReleasingLegalVerification	0.88
BG_ReleasingLegalVerification	0.39
IT_ReleasingVisuraForPartnership	0.90
IT_ReleasingAbridgedRegistrationCertificate	0.64

Table 4 Ordered *semsim* values between ofv_{RLV} and each ofv_{cs}

ofv_{cs}	semsim(ofv_{RLV}, ofv_{cs})
ofv_{cs3}	0.90
ofv_{cs1}	0.77
ofv_{cs2}	0.68
ofv_{cs4}	0.64

service *ReleasingVisura* is composed by three concrete services: cs_3, cs_1, and cs_2 that have a *semsim* degree equal to 0.90, 0.77 and 0.68, respectively.

GS(ReleasingVisura) = $\{cs_3, cs_1, cs_2\}$

This means that any concrete service in the GS is a proper concrete service for executing the *ReleasingVisura* abstract service.

Once the GS for the other abstract service, *Releasing Fiscal Verification*, has been identified, the final BPW, which is the actual executable workflow, is obtained by selecting one single CS from each GS and binding it to the corresponding abstract service. This last selection can be seen by the end user considering for instance the execution time and price of the concrete services.

6 Related Works

The main innovations in the use of semantic technologies in LD-Cast are represented by the semantic enrichment of the business process and its components and the semantic search and discovery method used to identify the services that are necessary to realize the business process. Accordingly, this section is organized in two subsections.

6.1 A Semantic Approach to Web Services and Business Processes

Semantic description of digital resources and web services, in particular, is one of the key issues of the semantic web activities. Several initiatives are addressing this field. In particular, among the most relevant we recall here WSMO, OWL-S, WSDL-S.

WSMO (Roman et al. 2006) defines an overall framework for Semantic Web Services consisting of four top level elements: Ontologies that provide the semantic terminology definitions used in all other element description as well as for the information interchanged in Web Service Usage, Goals for representing the objective that a client wants to achieve by using Web Services, semantic description of Web Services, and Mediators for resolving possibly occurring mismatches between elements that should interoperate.

OWL-S (Web Ontology Language for Web Services), part of the DAML Program, specifies a set of ontologies based on OWL to describe different aspects of a semantic Web service. There are three core ontologies, i.e., service profile, service model and grounding. Service profile presents "what a service does"; service model describes "how a service works"; service grounding supports "how to access it" via detailed specifications of message formats, protocols and so forth (normally expressed in WSDL). All the three ontologies are linked to the top-level concept Service, which serves as an organization point of reference for declaring Web services.

WSDL-S (Web Service Description Language-Semantics) (Akkiraju et al. 2005), developed by the Meteor-S group at the LSDIS Lab, proposes a mechanism to augment WSDL with semantics, in particular focusing on the services' functional descriptions. Based on the WSDL, WSDL-S has the advantage of attaching semantics building on existing Web services; in the meantime, it does not dictate a specific language for semantic description.

Among the previous initiatives, WSMO seems to be the most complete. In fact, its four top level elements represent the core building blocks for Semantic Web Service enabled systems. Nevertheless, the WSMO philosophy is towards a substitution of the current technologies and not for an integration of them. From this point of view, the Semantic approach defined in LD-CAST is more for a semantic enrichment of existing resources and does not impose any constraint on the technologies to be used for defining business processes, workflows or services.

6.2 Similarity Searching Algorithms

The problem of similarity reasoning has been widely addressed in the literature. The first problem concerns the similarity between pairs of concepts. Some methods to determine semantic similarity in a taxonomy are based on the so-called edge-counting approach (Rada et al. 1989) – that is, the shorter the path between nodes, the more similar the concepts associated with the nodes are. Unfortunately, this approach relies on the assumption that links in the taxonomy represent uniform distances that, in general, is not conform to the reality. For this reason, the approach based on the notion of information content has been proposed in the literature (Resnik 1995), which is independent of the path lengths of the hierarchy. Such an approach has been successively refined in (Lin et al. 1998), where a similarity measure showing a higher correlation with human judgments has been defined.

Other research results concern the similarity between two sets (or vectors) of concepts (Madhavan and Halevy 2003). In the literature the Dice (Frakes and Baeza-Yates 1992; Maarek et al. 1991) and Jaccard (1901) methods are often adopted in order to compare vectors of concepts. However, in both Dice and Jaccard, the concept of similarity is based on the set theoretic intersection and union of the two vectors. Therefore, the adoption of the mentioned approaches requires the exact match of the concepts of the compared vectors. With respect to Dice and Jaccard, our proposal allows a refinement of semantic similarity since the match of OFV components is based on the shared information content.

Finally, several methods for computing semantic similarity have been proposed in the literature (Madhavan and Halevy 2003; Maguitman et al. 2005), and some solutions proposed for the industrial sector, e.g., like the Semantic Tagging proposed by Centiare.

7 Conclusions

In this paper we presented the results obtained in the European project LD-CAST for what concerns the semantic solutions that have been developed. The objective was to achieve a smooth cooperation among European Chambers of Commerce and, furthermore, between the latter and the European SMEs by supporting a flexible construction of executable workflows. Such cooperation should be based on the flexible interoperability of the Information Systems that are used by the different CoC. The main problems encounter are caused by: (1) the above mentioned difference in the IS of the CoC; (2) the difference in the law and regulations that exists from country to country; (3) the fact that the level of automation of the provided business services is very different form one place to another, from a country to another; (4) the need to provide the requested services according to good practices, that may change in time; and finally (5) the possibility of customizing the service provision in accordance with the user profile and user needs.

We addressed the above problems by developing and ontology-based semantic suite capable of: (a) supporting the construction of a weighted reference ontology; (b) supporting the semantic annotation of business processes and concrete services offered to the user; and (c) providing a semantic search and discovery mechanism for the retrieval of concrete services. These three semantic services have been successfully applied in the LD-Cast platform and thoroughly tested in the different pilots. The final results have been very positive and the developed solutions proved that the semantic technologies have reached a good maturity level. Additional tests, and comparison of the *semsim* method with other similarity approaches, have been proposed by Missikoff et al. (2008). We are confident that soon the industrial adoption of semantic technologies will start to spread, and the research results of LD-Cast will produce an interesting result for eGovernment and for the European SMEs.

Acknowledgments We wish to address our warmest thankfulness to all the colleagues and partners of LD-Cast, since without their constructive cooperation these results would not have been achieved. In particular, we owe a special credit to Alessandra Catapano (ElsagDatamat) as the LD-CAST coordinator, to Antonio De Nicola (CNR-IASI) to Alessandro D'Atri and Paolo Spagnoletti (LUISS CeRSI), to Robert Woitsch and Vedran Hergovcic (BOC), to Nicola Christian Rinaldi and Valentina Di Michele (Retecamere).

References

R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M. Schmidt, A. Sheth, K. Verma, Web service semantics – WSDL-S (technical note), http://lsdis.cs.uga.edu/library/download/WSDL-S-V1.html, April 2005

ATHENA, D.A3.1, SoA on ontologies and the ontology authoring and management system, with ontology modelling language, ATHENA IP. Deliverable (2005)

T. Berners-Lee, J. Hendler, O. Lassila, The semantic web. Sci. Am. **284**(5), 34–43 (2001)

A. De Nicola, R. Navigli, M. Missikoff, A software engineering approach to ontology building. Inform. Syst. **34**(2), 258–275 (2009)

DSTC, IBM, Sandpiper software; Ontology definition metamodel, Revised submission to OMG, http://www.omg.org/docs/ad/05-01-01.pdf. Accessed 10 Jan 2005

W.B. Frakes, R. Baeza-Yates, *Information Retrieval, Data Structure and Algorithms* (Prentice Hall, Upper Saddle River, NJ, 1992)

T.R. Gruber, A translation approach to portable ontologies. Knowl. Acquis. **5**(2), 199–220 (1993)

P. Jaccard, Bulletin del la Société Vaudoise des Sciences Naturelles **37**, 241–272 (1901)

LD-CAST, D3.1. Common knowledge base specifications. Deliverable (2006)

D. Lin, in *An Information-Theoretic Definition of Similarity*, ed. by J.W. Shavlik. Proceedings of 15th the International Conference on Machine Learning, Madison, Wisconsin, USA (Morgan Kaufmann, San Francisco, 1998), pp. 296–304

Y.S. Maarek, D.M. Berry, G.E. Kaiser, An information retrieval approach for automatically constructing software libraries. IEEE Trans. Softw. Eng. **17**(8), 800–813 (1991)

J. Madhavan, A.Y. Halevy, Composing mappings among data sources. VLDB J. **2003**, 572–583 (2003)

A.G. Maguitman, F. Menczer, H. Roinestad, A. Vespignani, Algorithmic detection of semantic similarity. in *Proceedings of WWW'05 Conference*, May 2005, Chiba, Japan

M. Missikoff, F. Taglino, F. D'Antonio, Formalizing the OPAL eBusiness ontology design patterns with OWL. in *Proceedings of IESA07*, Madeira (Portugal) 26–27 March 2007

M. Missikoff, A. Formica, E. Pourabbas, F. Taglino, Weighted ontology for semantic search. in *Proceedings of ODBASE08* (to appear), Monterrey, Mexico, 11–13 November 2008

OWL-S (Web Ontology Language for Web Service) 1.2 Release, http://www.ai.sri.com/daml/services/owl-s/1.2/

L. Rada, V. Mili, E. Bicknell, M. Bletter, Development and application of a metric on semantic nets. IEEE Trans. Syst. Man Cybern. **19**(1), 17–30 (1989)

P. Resnik, Using information content to evaluate semantic similarity in a taxonomy. Proc. IJCAI **95**(1), 448–453 (1995)

D. Roman, H. Lausen, U. Keller, D2v1.3. Web service modelling ontology (WSMO). Deliverable, http://www.wsmo.org/TR/d2/v1.3/, October 2006

C.E. Shannon, A mathematical theory of communication. BLTJ **27**, 379–423, 623–656 (1948), http://mywikibiz.com/Semantic_tagging