4+1 Views for a Business Cooperation Framework Based on SOA

Anca Daniela Ionita, Monica Florea and Lucian Jelea

***Abstract* — The paper analyzes LD-CAST, a business cooperation framework for automated or semi-automated business services, realized by various processes that orchestrate late bound Web services. The 4+1 Views Architecture proves to be appropriate for the analysis and design of systems with complex objectives, like: user friendliness for multiple actors; maintainability and extensibility under business evolution conditions; flexibility for dynamic retrieval of service providers; performance traceability for software and users. The paper presents the correspondences established between use-case, process, logical, implementation and deployment views, such as to attain these objectives, and to hide the back-stage complexity for the business domain users. The solution is based on three pillars: Service Oriented Architecture (SOA), Business Process Modeling (BPM) and semantic Web. The three of them are also used for comparing our system with other similar works.**

***Index Terms*— e-Business, Service Oriented Architecture, UML Modeling, Web Service Orchestration**

I. Introduction

The new way of doing business stands in an increasing interaction inside the business community, by directly creating and maintaining partnerships between individual companies, community organizations and social entrepreneurs. Businesses of all sizes need to interact to each other, but also to public authorities [1] - in a friendly and transparent way, supported by new cooperation frameworks for Business to Government (B2G). The new alliances may be easier established in a homogenous environment, where social partners may interact with trade unions, regional and international governmental organizations and a wide variety of other civil society organizations and networks. Currently, the state of the art is still characterized by non-homogeneity related to legislation, terminology and procedures, but also to the technical environments, due to different data structures, communication protocols or semantics.

Manuscript received September 26, 2009. This work was supported in part by the European Community, under the FP6 IST Programme, FP6-2004-IST-4-26919 project (LD-CAST) and in part by CNCSIS, Romania, under the Grant IDEI 627/2009.

A.D. Ionita is with the Computers and Industrial Informatics Department, University "Politehnica" of Bucharest, Spl. Independentei 313, 060042, Bucharest, Romania (phone: 4-021-4029113; fax: 4-021-3181014 ; e-mail: Anca.Ionita@mag.pub.ro).

M. Florea is with SIVECO ROMANIA SA, Victoria Park, Sos. Bucuresti-Ploiesti 73-81, 013685, Bucharest, Romania (e-mail: Monica.Florea@siveco.ro).

L. Jelea is with SIVECO ROMANIA SA, Victoria Park, Sos. Bucuresti-Ploiesti 73-81, 013685, Bucharest, Romania (e-mail: Lucian.Jelea@siveco.ro).

Interoperability frameworks are being created for the cooperation and integration of public institutions, as well as for supporting business development, with the attention to address organizational, semantic and technical issues [2]. An important commonality of these platforms is that they are based on electronic services, which are developed for a large variety of applications, dedicated to citizens, businessmen, civil servants, medical staff or teachers. Service engineering involves cooperation of multiple tools and actors, not only for the front-end business specific functionality, but also for the back-end architecture, which attains a high complexity due to the requirements of managing processes and ontologies, maintaining smart service repositories, discovering services automatically and supporting security.

The present e-Business systems are confronted to a lot of challenges. Heterogeneous types of business services have to be integrated into seamless processes - at local, national, and Pan-European levels - eliminating the loose electronic integration of activities for Service Providers (SPs) from different European countries and regions. Documents, processes, and services have to be defined with no ambiguity. The automation of the request resolution inside SPs needs improvements, as well as the validation and monitoring of services and processes provided by multiple SPs. One has to attain flexibility in service/process reconfiguration and maintenance and to dynamically propagate changes of semantic definitions to registered services and processes. Personalized and easy-to-use interfaces are required for non-IT personnel. Automatic guidance is needed for all sorts of users, especially for non-specialist ones.

The scope of this paper is related to business development in the context of the enlarged Europe, in connection with the e-service frameworks existent at national levels. In order to implement trans-border processes, one has to surpass the difficulties related to:

- multi-actor environments that require multilateral agreements;
- non-homogeneity of security policies of the involved institutions;
- non-interference with the internal work of administrations, enterprises or institutions that deliver services;
- necessity to create strong multilingual facilities for customer-oriented services, processes and tools.

The technical problems arise from the necessity to create large-scale distributed systems that may be accessed through a unique, one-stop portal, hiding the existent non-homogeneity of procedures, document formats and legislations among European countries. Service Oriented

Architecture [3] offers possible solutions for these problems, by supporting atomic and composite services, eventually delivered by multiple cross-border service providers. Semantic non-homogeneity can be treated by defining ontologies [4] that characterize generic processes, abstract Web services and service providers from various countries, as well as common quality of service (QoS) notions, necessary for establishing service level agreements. Sometimes, correspondences between local ontologies (either regional or national) and central, European level ones [5] have to be defined, implemented and maintained.

The aim of LD-CAST system [6] presented in this paper was to support the development of private company initiatives, belonging either to private or public sectors, by cross border cooperation between European Chambers of Commerce (CCs). A one-stop portal at European level is dedicated to existent and new start-up enterprises, created by investors or entrepreneurs interested in an innovative line of business, based on trans-national development and growth. The semantic interoperability between different CCs acting as service providers is assured by an ontology containing common, core elements, which are then specialized on national-specific concepts. The technical interoperability stands in the framework developed for modeling, executing and monitoring business processes that seamlessly compose and localize services delivered by different institutions, from different countries, while respecting their specificities, procedures, as well as privacy and security rules and regulations.

The paper describes the LD-CAST system in chapters II and III, using UML 2 models [7], organized according to 4+1 Views Architecture [8]. It also presents the correlations between these views, outlining the interoperation between subsystems accessible to business domain users and those hidden to them, dedicated to actors responsible for the system configuration and maintenance (more details in chapters IV and V). Chapter VI treats configuration and maintenance issues and chapter VII analyzes some works that are related to LD-CAST approach, many of them also supported by European projects.

II. A Business Cooperation Framework based on SOA

The European market analysis highlights the strong presence of partnerships at trans-national level. In this context, the main advantages of supporting new interaction patterns are:

- creating new opportunities for training, placements, mentoring, and leadership development;
- improving operational efficiency;
- helping institutions to grasp new opportunities;
- increasing access to financial, technical and managerial resources;
- enhancing effectiveness and credibility.

Moreover, there are supplementary advantages that might be brought by a trustworthy system, offering a cooperation framework of cross-border services to enterprises and entrepreneurs:

- supplying a large spectrum of information related to the business support existent in various countries;
- using a common terminology;
- saving the client time, by delivering documents and/or services without the necessity to reach the local sites of chambers of commerce or other institutions;
- assuring a secure access.

To attain these goals, the system has to provide an organizational, semantic and technical interoperability, as specified in the European Interoperability Framework [2].

The scope of a business cooperation system covers activities like business set up, certification, company information, promotion, training, investment towards and from foreign countries, supplying socio- economical data and supplying access to on-line services. The difficulties to create an homogenous environment for them are due to the different degrees of technological development of public administrations and to the fact that, currently, most of the services provided by the chambers of commerce are delivered at the desk, while only few on-line services are available.

One of the important tasks of LD-CAST project was to create unifying concepts and tools regarding various cooperation aspects, like cooperative business, compatibility of statutes, communication protocols and common repositories. The compatibility from the point of view of terminology was also a key issue; hence, the multilingual interface of some tools and the creation of a common business glossary, such as to avoid lexical ambiguities. The technological incompatibilities were smoothened by choosing service oriented architecture.

To take advantage of the business cooperation system, an entrepreneur or an SME would have to get registered at one of the local agencies, which act as connection points and as identity providers, and have responsibilities regarding the access of the local users, the payment and the provision of information regarding the available services. After registration, they implicitly become LD-CAST End-Users, entitled to access the following functionalities:

- browsing, selecting and requesting for complex business services;
- making choices related to the preferred configurations, particularly related to cost and time;
- delivering and receiving documents;
- monitoring the execution of business processes;
- communicating with the platform administrator.

In order to offer this support for the business parts, the framework uses Web services delivered by external service providers, and orchestrate them according to specific processes, in order to execute the requested business service. Therefore, other important functionalities stand in:

- defining and publishing Web services deployed by service provider registered at LD-CAST;
- defining and publishing processes that base their execution on the orchestration of the available Web services.

One of the most important requirements of LD-CAST was to map process activities on actual services at run-time, not at design time, for increasing the system flexibility. For this purpose, both services and processes are annotated according to a common business ontology, and matchmaking can be performed dynamically.

For the administration of the framework, the main functionalities are:

- monitoring system performance and quality of service of the service providers;
- fulfilling security requirements.

This subchapter has briefly the requirements of LD-CAST informally; the following one is applying the 4+1 Views Architecture to describe the way the system was developed.

III. 4+1 Views for LD-CAST Framework

Viewpoint-oriented approaches have often been used in requirements engineering, to be able to treat the needs of various stakeholders [9]. Moreover, they have been useful for designing the architecture, as it is the case of the approach of Philippe Kruchten [10], applied in this paper, and also for developing based on separation of concerns and Aspect Oriented Software Development [11].

LD-CAST system is presented below, according to the following 4+1 views (see Fig. 1):

1) *The Logical View* – representing structural elements, levels of abstraction, separation of concerns and frequently using object-oriented models;
2) *The Process View* – treating the execution aspects of the system: the interaction patterns and the concurrency;
3) *The Implementation View* – showing the decomposition in subsystems used in the development environment and originally called development view;
4) *The Deployment View* – outlining the distribution of software components on physical nodes, either computers or processors; it was originally named physical view.

+1) *The Use Case View* – illustrating the use cases or scenarios dedicated to multiple actors and further involving dedicated languages and tools, which are described in other views.

The particularities of this system are related to SOA – affecting the deployment. Moreover, the presence of the executable business processes gives a larger importance to the Process View and to the correlations between it and the others. Due to the generic character of the "4+1" view model, the paper uses UML diagrams as concrete notations.

A. The Use Case View

LD-CAST is mainly dedicated to serve entrepreneurs interests to obtain electronic business services, in order to save their time and money and to attain confidence related to unknown chambers of commerce and potential partners from abroad. Any *Guest* is able to obtain detailed information related to business cooperation, but one has to register and become an *End-User,* in order to get advantage of the business services published within the framework (see the use case diagram in Fig. 2).

However, LD-CAST system had to assure the dynamic discovery of Web services, taking into account the client preferences, the common descriptions for services and processes, and the respect of the subsidiary principle, by allowing each chamber of commerce to follow its specific procedures. In order to acquire these requirements, the system needs specialists with various competences for its configuration and maintenance.

Business Process (BP) Designer is in charge of modeling specific processes, whose activities are to be implemented by Web services (see Fig. 3).

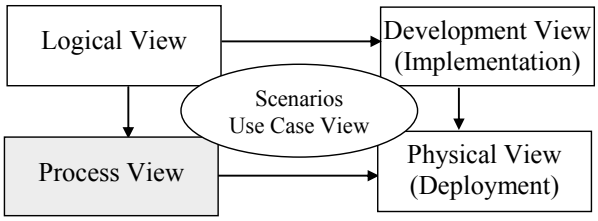

Fig. 1 Correspondences between the 4+1 Views

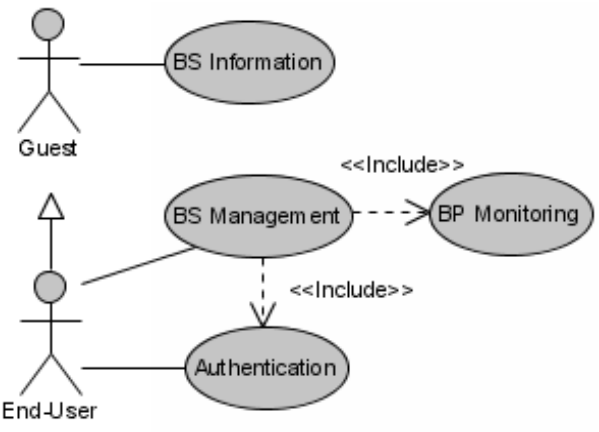

Fig. 2 Use case diagram with non-specialist actors [12]

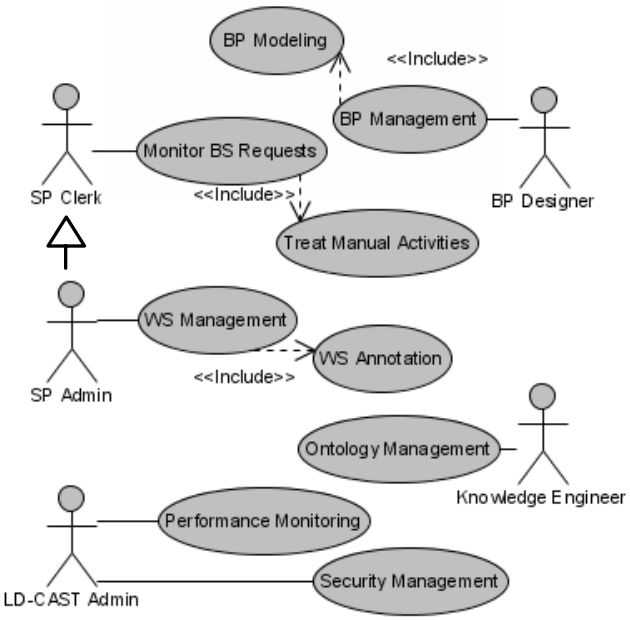

Fig. 3 Use case diagram with actors involved in LD-CAST configuration and monitoring [12]

The late mapping of activities on concrete services is realized through the concepts and relationships of the business ontology, created and managed by the *Knowledge Engineer.*

Thus, the process is executed with actual Web services, selected from a repository where they had been published by persons playing the role of *Service Provider Administrator.*

The security and integrity of LD-CAST are assured by an *LD-CAST Administrator*, who can also monitor the clients

and the system performance.

Even if the system was conceived to automate the delivery of business services, the practice revealed that some of the tasks still have to be performed manually; therefore, a *Service Provider Clerk* was introduced, who can also monitor all the business service requests assigned to its service provider.

B. The Logical View

Each subsystem of LD-CAST has a complex data model. For the purpose of this paper, let us see the logical point of view standing behind the execution of a business service, which is a key issue for this framework. There are three layers [13], corresponding to different levels of abstraction and different technologies (see Fig. 4):

- Modeling Layer
- Abstract Execution Layer
- Concrete Execution Layer.

The *Modeling Layer* is situated at the highest level of abstraction and contains the following classes:

- *Business Service* (BS) – representing an intangible product achieved to fulfill a need or a goal; it is characterized by the defined inputs and outputs and by the quality of service;
- *Business Process* (BP) – a process that is associated to a business service and may render it executable;
- *Activity* - is a part of the business process that may be undertaken automatically or manually by an entrepreneur, a commercial enterprise or a chamber of commerce.

LD-CAST has identified the following business services that can be delivered by the chambers of commerce: Partner search, Company legal verification, Company fiscal verification, Technical and quality standard verification, Agreement, Registration, Mediation, Arbitration and Supply subcontract. However, each of them may be delivered by different chambers, from different countries, involving a 1-to-many correspondence between a business service and its business processes, defined by a domain expert playing the role of BP Designer.

In order to automate these processes, they have to be transformed into executable workflows. As LD-CAST flexibility requirement imposes a late binding of Web services, there are actually two layers of execution.

The *Abstract Execution Layer* contains the classes:

- *Abstract Workflow* – a sequence of abstract services, corresponding to a business process, but introducing technical details, to allow the automatic generation of a Concrete Workflow;
- *Abstract Service* - described by its functionality, input and output; it is not directly executable, because it does not refer to a specific service implementation (a concrete Web service); it is defined at design time and it represents a set of Web services with the same functionality, input and output .

The *Concrete Execution Layer* contains the following 2 classes:

- *Concrete Workflow* - a workflow for which each abstract service has been replaced with a correspondent Web service;
- *Web Service* - an executable service representing a concrete service having the same functionality, input and output as the correspondent abstract service.

The concrete workflow is determined at run time, by semantically searching the available concrete Web services that may implement the abstract activities. The service discovery introduces a first level of flexibility in the traditional way of modeling and executing processes: the business service is described in terms of abstract activities, ignoring the way a specific activity is implemented; there is no actual reference to the concrete service that realizes that activity. The Business Process Designer actually designs a template for the business process, which will be rendered executable at run time, after being requested by the End-User.

C. The Process View

The process view of LD-CAST is rather complex; there are two kinds of processes that have to be descried in this view:

- *scenario processes* – corresponding to the use cases associated to the system actors and represented as activity, sequence or communication diagrams; they have a descriptive role in the system analysis and design.

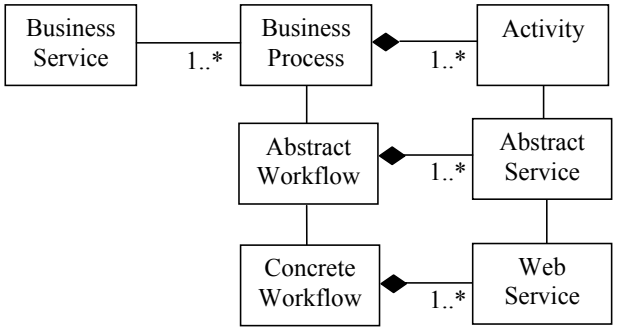

Fig. 4 The logical view with the execution layers

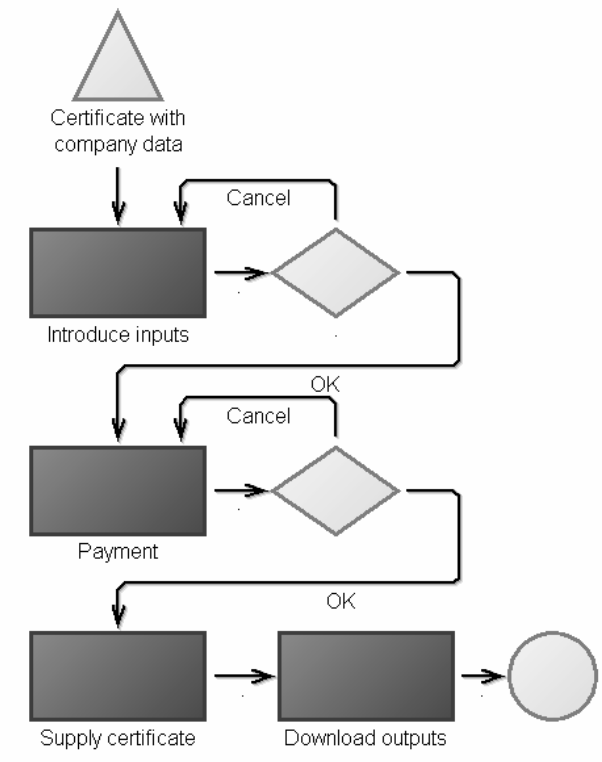

Fig. 5 Example of business process: Certificate with company data

- *executable business processes* – used for configuring the system; the are represented with a specific language at the level of the Modeling Layer and then with BPEL at the Abstract and Concrete Execution Layers, to be able to be transformed into an executable workflow .

Let us analyze in this paper the example of business process from Fig. 5, which is defined for supplying a certificate with legal or fiscal data related to a certain company. The legal verification may contain, in terms of the country, the certificate of existence, information about the joint stock, the corporate form, the business history, or a criminal report to know if the company has been involved in criminal acts in the past. The fiscal verification may contain a report about the main financial indicators, the full balance sheet and the regular pension contribution certificate.

For such processes, some activities are executed automatically; others require the contribution of various actors. In our example, the End-Users interact for introducing inputs that personalize the request, for making the payment and for downloading the results. Ideally, the certificate is generated automatically and supplied through a Web service; currently, for some chambers of commerce it has to be supplied by the Service Provider Clerk.

D. The Implementation View

The LD-CAST architecture is service oriented and all the connections between its subsystems, depicted in Fig. 6, are realized through Web services, using SOAP [14]. The subsystems accessed by each of the actors mentioned above are presented as follows.

The Guest and the End-User work with the *Run Time Portal* for obtaining information and business services from LD-CAST. The portal is implemented with JEM (JBoss Enterprise Middleware) [15] and performs its tasks dynamically, by interacting with other subsystems.

The Business Process Designer uses the *Business Process Modeler* (BP Modeler) subsystem, for defining process models with ADOeGov [16] modeling language, which allows process activities to be linked with abstract workflows

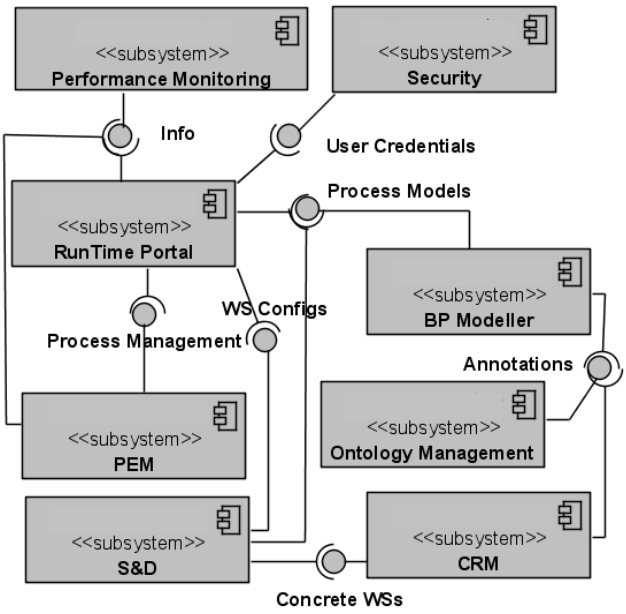

Fig. 6 LD-CAST subsystems and their connectors [12]

written in BPEL4WS (Business Process Execution Language for Web Services) [17].

The Knowledge Engineer uses the *Ontology Management* subsystem to create the reference business ontology, using OPAL (Object, Process, Actor Modeling Language) [18]. An ontology gathers a set of concepts concerning entities, attributes, processes, together with their definitions and inter-relationships. The reference ontology, related to the chambers of commerce domain of expertise, is used for semantically annotating Business Processes and Web Services, i.e. for attaching an expression built in terms of a reference ontology. The semantic annotation is then used for enabling the searching capabilities, in order to retrieve the needed resources.

The Service Provider Administrator accesses the *Concrete Resource Management* (CRM) subsystem, in order to store information about the service provider, as well as to register, annotate and publish the delivered Web services. Manual services delivered though LD-CAST by a clerk of the service provider (not automatically) - are also registered at CRM and consequently, all activities of a business process are treated in a uniform way. Information is stored in a repository that may be accessed by the *Search and Discovery* (S&D) subsystem, which is able to perform a semantic matchmaking between the set of available Web services and the abstract services corresponding to a requested business process. The configuration of actual Web services or manual activities that implement the workflow activities is not fixed in advance, but established through a selection made by the End-User at run time, according to their registered cost and time. Thus, one supports a late binding and the generation of concrete BPEL workflows, whose execution is performed by the *Process Execution Management* (PEM) subsystem, using the ActiveBPEL engine. S&D and PEM do not interact directly with any of the system actors.

The *Security* subsystem supports a federated identity solution for authentication and authorization, based on Shibboleth [19]. LD-CAST relies on the definition of a circle of trust with its local agencies - where End-Users are registered - while the management of other groups of users is performed directly, by the LD-CAST Administrator; moreover, he or she can also verify the system and the user efficiency, with the help of the *Performance Monitoring* subsystem, which collects logs from other subsystems and supplies a centralized monitoring console for the desired criteria.

E. The Deployment View

A primary reason for choosing SOA was the high degree of geographical distribution of LD-CAST, which is twofold. On one side, there is the deployment of the subsystems depicted in Fig. 6, which constitute the *LD-CAST Core* and for which the minimum number of nodes is 3, corresponding to the following platforms (see Fig. 7):

1) The *Modeling platform* is dedicated to system set up and configuration, and can be accessed by actors like: BP Designers, Knowledge Engineers, and SP Administrators. The following subsystems are part of this platform: Business Process Modeler, Ontology

Management and Concrete Resource Management.

2) The *Execution platform* supports the automatic execution of business services, by integrating the Run Time Portal, the Process Execution Management subsystem and the Search and Discovery one. Even if the portal is mainly dedicated to End-Users, it also allows the LD-CAST Administrator to manage its content, and supports the delivery of non-automatic services by the Service Provider Clerks, after introducing the proper credentials.
3) The *Administration Platform* is used by the LD-CAST Administrator to manage LD-CAST users (by interacting with Local Agencies - for End-Users - and directly - for the others); it also controls the efficiency of the system operation, using the Performance Monitoring subsystem.

Actually, the prototype was deployed on more execution nodes, in respect with responsibilities assumed by different members of the project consortium for the development and maintenance of various subsystems, even inside the same platform. The SOA architecture allows this flexibility, in terms of optimizing the communication times of its connectors.

On the other side, there are also two other kinds of entities, with their correspondent deployment nodes (see Fig. 7):

- The *Local Agencies*: responsible for the payment and for the local management of users that have access to the LD-CAST Core services; one becomes an End-User of LD-CAST and is able to request business services only after registering and identifying at a local agency, following its internal procedure; after that, the agency supplies its identity in the federated network.
- The *Service Providers*: which provide manual and Web services that will be orchestrated by LD-CAST Core, in order to satisfy an End-User request; they register and publish their services using the LD-CAST Core subsystems, through the SP Administrators.

IV. Correspondences between Views

Typically, the use cases make the link between all the other views and, even if considered redundant information in the 4+1 Views model, they are the first to be described. Thus, they clarify the usage of the system, especially when it is accessed by multiple actors and it has a complex structure.

The Logical View may be vertically divided in complex data models for each subsystem, but also horizontally, into a layered architecture, as descried in Fig. 7. Different actors of the system are aware of different classes and different layers of abstraction. The Guest can only see the business services; the End-User can also see the names of the business processes, plus some information related to the concrete Web services that may be used for the execution, such as to be able to make a choice. However, the BP Designer can define all the details related to the business processes and their associated abstract workflows. The SP Admin accesses the lowest level of execution, because it registers the concrete Web services in the repository.

The Process View is very complex in LD-CAST, for two reasons:

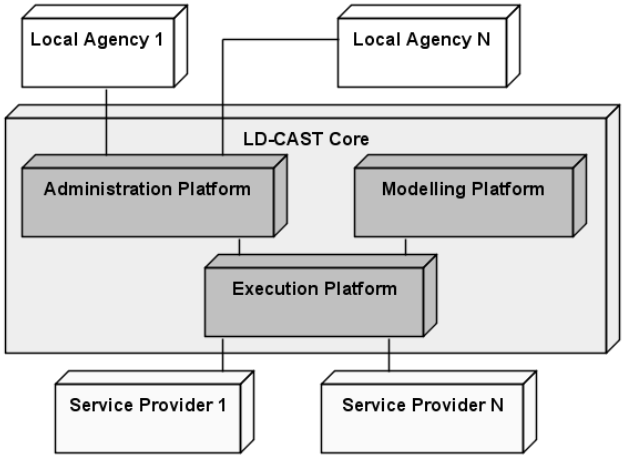

Fig. 7 LD-CAST deployment diagram [12]

- due to the existence of 7 different kinds of actors, each one with its rights and roles for using, configuring and maintaining the system;
- due to fact that business services are explicitly represented as processes, executed automatically and monitored by the framework.

The execution of a business process involves the End-User, but also the SP Clerk. Moreover, the performance of the End-User and of the Web services is monitored by the LD-CAST Administrator. More details related to what happens in the back stage of this execution are given in the case study from subchapter VI.

The Implementation View outlines that the subsystems have been specifically designed for different actors (see subchapter III). The non-professional actors can only access the Run-Time Portal, while the professional ones have to model and administer the system, so they are associated to the correspondent platforms from the Deployment View. The LD-CAST Admin works with Performance Monitoring and Security subsystems. The BP Designer configures the system with the BP Modeler. The Knowledge Engineer maintains the ontology with the Ontology Management subsystem and the SP Admin accesses the Concrete Resource Manager subsystem. Moreover, the SP Clerk, who is a business actor, not a technical one, also does his or her job through the portal, strictly contributing to the execution part.

V. Case Study for the "Business Service Management" Scenario

For the purpose of the paper, one selected a typical scenario, correspondent to the *Business Service Management* use case (see Fig.3) with the correspondent activity diagram represented in Fig. 8.

First, the End-User has to select the desired business service (e.g. *Company Legal Verification*) and the portal displays a list of processes that have been published for it, reflecting specificities characteristic for a certain country or even for a certain chamber of commerce. After selecting a process (e.g. *Release Ordinary Visura*, from Italy) the system searches the configurations of concrete Web services that can be used for the actual execution and gives detailed information about them, as shown in Fig. 9.

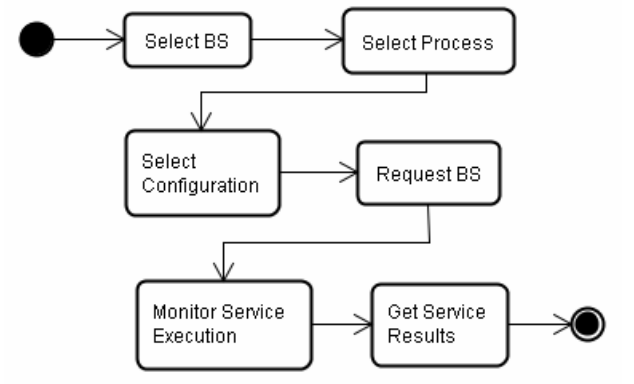

Fig. 8 Activity diagram for Business Service Management

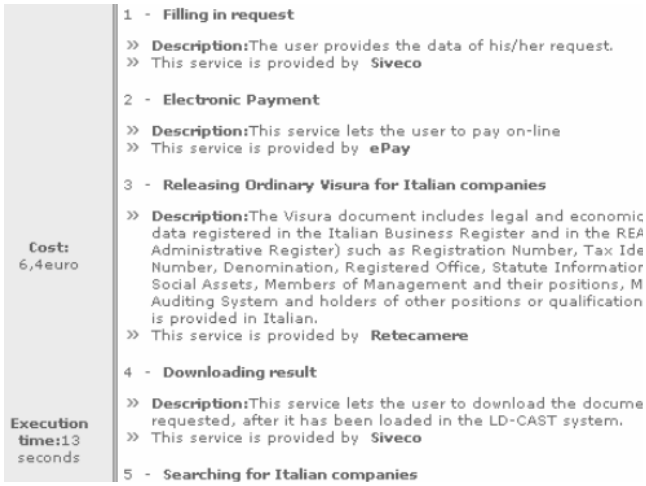

Fig. 9 Details related to a configuration of concrete services

The End-User selects the preferred configuration and starts the execution of that business process, then monitors it, by actively getting involved in certain activities. At the end, it is possible to download the expected results, which have been stored in a repository of the one-stop portal.

The case study analyzes the correspondences between elements of different views for the following activities:

- Select Business Service,
- Select Process,
- Select Configuration and
- Monitor Service Execution.

A. *The Activity "Select Business Service"*

For the activity "Select Business Service", the actor involved is the End-User, while the logical concept is Business Service, situated in Fig. 4 at the highest level of abstraction.

There are 2 correspondences to the Implementation View:

- the Run Time Portal - which displays the list of business services and the description of the selected one (see Fig. 10) and
- the Business Process Modeler - from whom this information is actually taken, as the content of the portal is created dynamically.

Thus, this simple activity involves a process that goes across the Execution and the Modeling Platforms; the communication is performed with Web services.

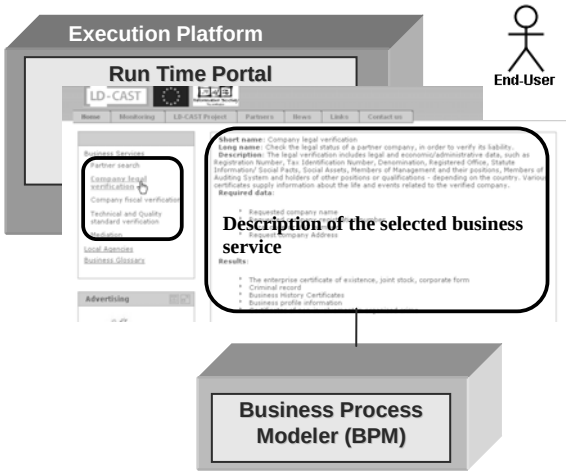

Fig. 10 Mappings for the activity "Select Business Service"

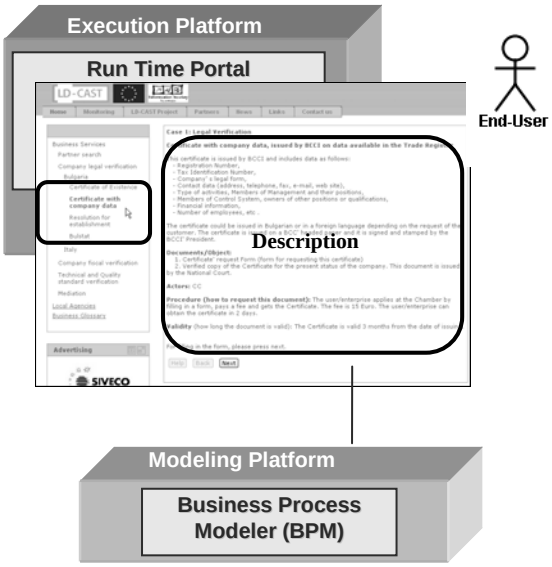

Fig. 11 Mappings for the activity "Select Process"

B. *The Activity "Select Process"*

The tree from the left area of the portal can be expanded in order to display the names of all the objects of type Business Process associated to a business service (see Fig. 11); note that one still remains at the highest level of abstraction in the Logical View. The correspondences to the Implementation and Deployment Views are similar to the previous activity.

C. *The Activity "Select Configuration"*

Even if, from the point of view of the End-User, this activity is not different from the previous ones, it involves more complex interactions in the backstage, in order to allow the system flexibility and to take into account the user preferences.

For the business process selected at the previous step, a correspondent abstract workflow has been previously implemented. It represents the control flow, with the sequencing of operations, but without concrete implementations of the required activities. For creating a concrete workflow and getting at the third level of execution from the Logical View, the user has to choose the concrete services to which the system will bind the activities. A

configuration consists in a set of Web services that are registered at CRM and match to the abstract activities from the point of view of inputs, outputs and semantics. Thus, Run-Time Portal interacts with Search and Discovery, which uses a semantic matchmaking mechanism, based on semantic annotations performed on concrete services from CRM and business processes from BPM (see Fig. 12).

Since there can be more than one Web service suitable for a given activity, the semantic search will return a set of Web services. Thus, given a business process, there can be more combinations of concrete Web services (configurations) suitable for being executed.

D. The Activity "Monitor Service Execution"

The End-User is not aware of the execution of Web services, nor of the workflows that orchestrate them. He or she needs a console (given by the Run-Time Portal) to monitor business service requests, checking their progress and status. The work is done at a high level of abstraction – that of the business process, modelled with BPM.

Moreover, the End-User has to interact with the system during the workflow execution, for instance for filling-in forms, uploading documents, downloading results, or starting manual activities.

The End-User is allowed to see information related to the processes that he or she had instantiated, whose states may be:

- Executing,
- Finished or
- Errors.

For each process it is possible to see the list of activities, whose states may be:

- Not Started,
- Started,
- Finished,
- Errors.

Everything is deduced from the process state stored by the workflow engine of PEM subsystem (see Fig. 13).

VI. Case Study for the Process "Certificate with Company Data"

Even if scenarios are generally used for "putting it all together", the Process View can also clarify the necessity of various subsystems used for the implementation, as well as of different nodes were they are deployed. This case study is oriented on the executable business process from Fig. 5, intending to show what is behind two of its activities: Payment and Supply Certificate. This process model, realized with BPM, was necessary because executable workflows are too complicated to be used by business experts and can not be designed by them directly. Therefore, LD-CAST used a metamodel of the process management domain and a friendly user interface. The studied activities can also be seen in Fig. 14, on the top level.

The abstract workflow corresponding to this piece of model is also depicted at the second level, as represented with a BPEL Editor (see Fig. 14). For each process activity there is one correspondent abstract service (an activity that should call a Web service) and several concrete Web services that match (see the third level from Fig. 14).

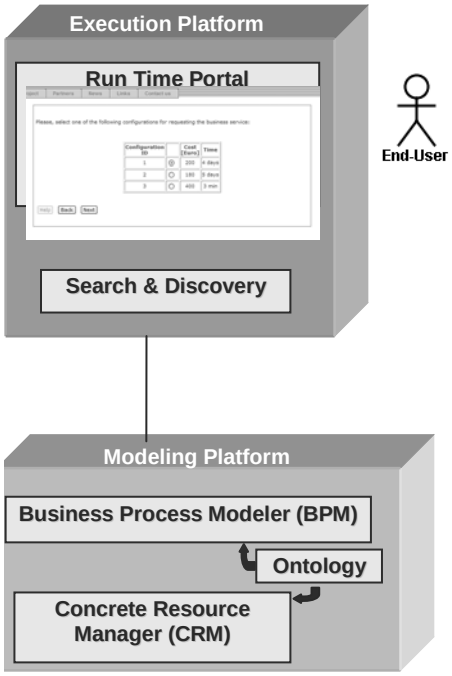

Fig. 12 Mappings for the activity "Select Configuration"

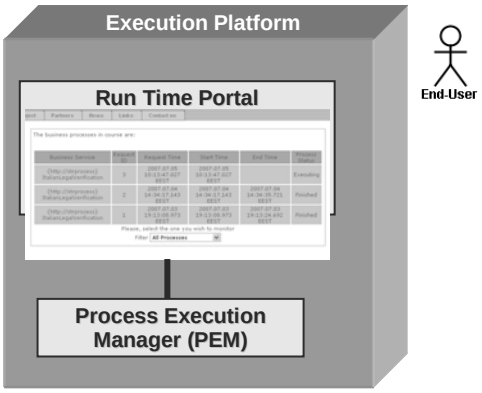

Fig. 13 Mappings for Monitor Service Execution Activity

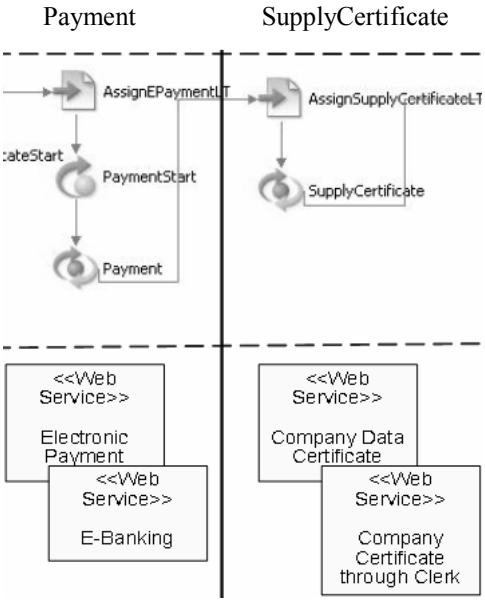

Fig. 14 Mappings for Payment and Supply Certificate Activities

This means that, for each one of the activities Payment and SupplyCertificate, there are 2 Web services registered at Concrete Resource Manager that may be called for the execution. For example, the activity SupplyCertificate may be mapped on 2 Web services (registered at CRM):

- *Company Certificate through Clerk* – a Web service supporting a specific interaction with SP Clerk, in order to obtain the results;
- *Company Data Certificate* – a Web service that automatically supplies a certificate.

For the former, the requested certificate requires information that can not be automatically extracted from a database, as it often happens in real life scenarios; providing the service implies interaction with a clerk who had the possibility of producing and certifying the document, and then uploading it to the system to be later downloaded by the user. LD-CAST system registers this service at CRM and uses it in the execution of workflows as an alternative to the latter, which has a similar result, but obtains it automatically

Furthermore, the system may be extended by plugging in new services, which have the same inputs and outputs and are annotated with the same concept from the ontology. Thus, building and then using an ontology related to the business service domain plays a fundamental role. The ontology does not only guarantee the mapping between the high level process activities and the Web services actually executing them, but it also allows business experts to continue working with meaningful concepts, abstracting from the technical issues.

This process was used for testing the integration between the Modeling Platform and the Execution Platform. The process is published in the Modeling Platform and is retrieved from the Execution Platform, by the Run-Time Portal, after the End-User selects it. It has a correspondent abstract workflow – stored in Process Execution Manager – which has to be transformed into a concrete workflow, such as to be executed on the Workflow Engine from PEM. For this purpose, PEM interacts with Search and Discovery subsystem, which gives a list of possible configurations of Web services that might implement the abstract workflow; the example involves 4 possible configurations. The End-User selects one of these configurations through the Portal and then PEM transforms the abstract workflow into a concrete one and starts its execution.

VII. Configuration and Maintenance

For being able to support a typical scenario, as described above, the Run Time Portal requires services from:

- the Security subsystem, to get the user credentials;
- the BP Modeler, to get the list of business services and the descriptions of the processes published for them (an extract of the exchanged information in XML format is given in the Appendix);
- the Process Execution Manager, in order to start and monitor the execution of a concrete workflow, selected by the End-User;
- the Search and Discovery subsystem, to get the list of configurations available for building a concrete workflow;
- the Performance Monitoring subsystem, in order to supply information about End-Users operation and statistics related to the requested configurations, business processes and business services.

The Run Time Portal has the biggest coupling in the system.

LD-CAST has to be properly configured in order to allow a coherent operation. On one side, for the desired business service, at least one process has to be modeled with BP Modeler, has to be annotated with concepts of the Ontology and has to be associated with an abstract workflow. On the other side, for assuring the executability, for each abstract activity there should be at least one concrete Web service that has been registered at CRM, annotated similarly as the activity and published. Then, the business process can also be published, in terms of existing at least one concrete workflow corresponding to it.

LD-CAST has been designed as a system that is highly adaptable to change, as the system has to evolve in the rhythm of the business domain. Some important facilities for maintaining LD-CAST are: adding new concepts in the ontology (performed by the Knowledge Engineer); defining new processes (by the BP Designer); registering new Web services (by the SP Administrator); accepting new local agencies and service providers (by the LD-CAST Administrator) [20]. The difficulty stands in keeping the system coherent, under the conditions of multiple subsystems and multiple actors involved in the extension of the system. At the same time, the End-User must be oblivious to the existence of all the subsystems behind, but should notice the benefits gained through the potential of increasing the number of available business services and configurations corresponding to them.

VIII. Related Works

A. Works based on Service Oriented Architecture

The emerging electronic society has forced the development of SOA systems [21], where electronic services may be provided by: governments, business companies or even by consumers or individuals, and may be consumed by: other governments, business parts, consumers or by employees and citizens. Taking e-Business as example, information technology got involved in a large variety of business functions, like sourcing and procurement, design and production, logistics and distribution, marketing and sales, customer service [22]. Besides, a lot of European projects have adopted architectures oriented on services; some examples are commented below.

SOA4ALL [23] aims to develop a domain-independent service delivery platform, seeing SOA as "the emerging dominant paradigm for application development". The integration is based on a Distributed Service Buss and the basic functionalities and activities are supported by platform services. On this basis, the final users provide Web services and light-weight processes and are able to monitor their whole lifecycle.

BREIN [24] supports the realization of virtual organizations for business-to-business collaborations, using a Modeling Framework based on SOA. Some important contributions are related to an enhanced management of service level agreements [25], using semantic descriptions,

and to automatic reactions to service failures for increasing the reliability of consumers.

ALIS goal is to develop an innovative system in order to provide European citizens and private companies with a transparent, fast, secure and reliable access to the European Legal Knowledge. The main service-provision component in its architecture is the Legal Reasoning Engine (LRE) which is deployed on a service-oriented architecture for assuring the accessibility within e-Societies [26].

NEXOF-RA [27] is a strategic project building a Reference Architecture for a generic open platform facilitating the collaboration between service providers and third parties.

Generally, a new concern is to create user-centric instead of program-centric SOA, such as to enable a large deployment of services. Web 2.0 provides new methodologies regarding the way user interacts with Web applications. The semantic enabled services solve a great deal of the back-end problems, defining formally the meaning of information and services, but other complications may result at different levels.

A difficulty encountered in LD-CAST was that the user interface cannot be designed statically, but has to be adapted in respect with the input and output formats of whatever new services registered to the system when it evolves. Moreover, service orchestration [28] leads to the creation of composed services, involving composed input and output forms, whose configurations is not known in advance, but is decided by the End-User, at run time.

A similar requirement can be observed in SEGOF project [29], which proposes a way to generate e-Government forms semi-automatically, based on semantic descriptions of public administration services and their business rules. The project focuses on automatically identifying relevant input for a Public Administration Service based on the semantic description of the service and its business rules, which must be applied to create the particular results. According to the relevant input, a web-form is generated to gather the needed information from citizen. This input is interactively checked against the procedure business rules. If the provided data is correct and conforms to the ontology restrictions, it is represented as XML and RDF and can be consumed by any application supporting the data interchange standard, including semantic Web services.

The guiding principles for the future front-end technologies are: empowering end-users to select, create, compose and customize their own services; detecting contextual information to adapt seamlessly to each situation; using End-Users knowledge to improve service front-ends; enabling collaboration between employees, customers, developers and providers [30]. One example may be FAST project [31], which provides a visual programming environment, supporting composition of services and definition of business processes from a top-down user-centric perspective, while in the back-end there are semantic Web services.

B. The Role of Semantic Technologies

Loose coupling can be considered one of the main requirements in the development and implementation of a service-oriented system. The main motivations in developing a loosely coupled application are the benefits of extendibility, reusability, interoperability and modular maintenance. From a technological point of view, the challenges of these requirements are usually overcome by enhancing SOA with ontology management systems.

In particular, for avoiding predefined mappings between abstract and concrete services and for taking into account quality criteria and user choices, SOA generally uses Semantic Web Services. The role of semantic Web for assuring interoperability inside highly distributed systems has also been outlined by other European projects.

BRITE project [32] uses ontologies to mitigate semantic heterogeneity of information, data and processes, supporting proactive information delivery and coping with multiple levels of formality on information and data levels.

SAKE [33] adopted a semantic-based change management inside their collaborative and knowledge space dedicated to public administrations.

SEEMP [5] investigated interoperability between non-homogenous e-Government systems for the employment sector, pertaining to different countries among Europe. Their solution for resolving heterogeneity was to convert all the XML semantic content in terms of a reference ontology, shared by all partners, and then to lower it back to an XML corresponding to a different local ontology. Technologically, this is achieved using the WSMX Data Mediation [34].

SemanticGov project [35] utilizes the WSMO framework (implementing WSML ontologies) and Governance Enterprise Architecture (GEA) models for functionally restructuring public administration services and enhancing interoperability.

C. Executing and Monitoring Business Processes

Another important issue in LD-CAST was that business processes have to be defined at a high level of abstraction, by business domain experts, but they have to be rendered executable, in order to assure the proper user friendliness of the system.

A possible solution, adopted for Apel [36], is to create a Domain Specific Language (DSL) for defining process models and to use the DSL interpreter as execution engine.

Another solution (adopted in LD-CAST also) is to use a specific language for defining business processes and then to translate it into executable BPEL workflows, for which execution engines already exist. In this case, the problem is that automatic generation of BPEL code encounters a number of limitations: different process engines require different workflow code, and domain-specific models need complex graph transformation algorithms, for coming from a graph-based to a block-based structure.

Important achievements were obtained in AgilPro project [37], where one uses a generation framework for a component-based design, which fosters reuse and composition for parts of generation solutions. The generation framework aims to enable people with no or little experience in programming or workflow technology, to generate executable processes from models defined at a higher level of abstraction.

IX. Conclusion

A main problem in business-to-business applications at a large scale, for example at European level, is the non-homogeneity encountered both at conceptual and at technical levels. This concerns terminology, legislation, rules, applications, user interfaces and, last but not least, languages used in the daily activities. Service Oriented Architecture aims to reduce the collaboration problems and to offer a trustful environment for all actors involved.

LD-CAST system was designed to offer a common framework for business services, realized by automated or semi-automated processes, obtained by orchestrating late bound Web services. One of the big challenges was to offer flexibility in respect with the clients preferences, the national practices and the diverse terminologies. The idea was that End-Users, even if registered at Local Agencies that may be geographically distributed all around Europe, should retrieve a friendly system and should pass seamlessly through all the necessary steps, for treating their business service requests, without perceiving the complexity behind.

By describing the Use Case, Process, Logical, Implementation and Deployment Views of LD-CAST, the paper outlined how SOA, semantic Web and Business Process Modeling have been combined for attaining requirements that have different priorities for different types of actors: user friendliness and security for the End-User, maintainability for the Knowledge Engineer, extensibility for the Business Process Designer and Service Provider Administrator, and traceability for the LD-CAST Administrator.

Moreover, the paper showed the system integration by starting from a given use case and then analyzing the mappings on all the other views: Logical, Process, Implementation and Deployment ones. As SOA also involves workflows for orchestrating Web services, the process view has an important influence on the other views, imposing: logical layers; the presence of specific components for modeling, executing and monitoring workflows; communication bridges between computation nodes.

Appendix

One presents below a part of the XML description for the *Company Legal Verification* business service, as transmitted by the *BP Modeler* subsystem to the *Run Time Portal*, which subsequently displays information to *Guests* and *End-Users*.

<ATTRIBUTE name="PortalGroup" type="STRING"> Company Legal Verification,Italy</ATTRIBUTE>

<ATTRIBUTE name="gdShortName" type= "LONG STRING"> Company legal verification </ATTRIBUTE>

<ATTRIBUTE name="gdLongName" type="LONG STRING"> Check the legal status of a partner company, in order to verify its keeping of the operational current legal rules.</ATTRIBUTE>

<ATTRIBUTE name="gdDescription" type="LONG STRING"> The legal verification includes legal and economic/administrative data, such as Registration Number, Tax Identification Number, Denomination, Registered Office, Statute Information/ Social Pacts, Social Assets, Members of Management and their positions, Members of Auditing System and holders of other positions or qualifications - depending on the country. Various certificates supply information about the life and events related to the verified company.</ATTRIBUTE>

<ATTRIBUTE name="gdResults" type="LONG STRING"> The enterprise certificate of existence, joint stock, corporate form; Criminal record; Business History Certificates; Business profile information; Certificates of non-involvement in organized crime.</ATTRIBUTE>

<ATTRIBUTE name="gdNote" type="LONGSTRING"> The inputs and outputs differ from one country to another and also depend on the organization that supplies the service. The exact information may be obtained after registration to LD-CAST, through of its Local Agencies.</ATTRIBUTE>

Acknowledgment

We would like to thank to the LD-CAST Consortium, including both technical partners and chambers of commerce, for the experience we have gained together related to the interoperation between people and software, as well as related to the delegation of responsibilities to multiple types of users and to developing the correspondent tools. We would also like to express special thanks to ELSAGDATAMAT (Rome, Italy) for their support during the system development.

References

[1] Galant, V., "Blending E-Commerce Theory and Application", *IEEE Distributed Systems Online*, vol. 6, no. 1., 2005

[2] European Interoperability Framework for Pan-European eGovernment Services. Version 1.0. *European Communities, 2004* Available: http://europa.eu.int/idabc

[3] B. Fitzgerald and C. M. Olsson Eds. "The software and services challenge. Contribution to the preparation of the technology pillar", *Software, Grids, Security and Dependability of the 7th Framework Programme*. Ver 1.1. 30 Jan, 2006.

[4] D. Gašević, D. Djurić, V.Devedžić, *Model Driven Architecture and Ontology Development*, Springer-Verlag, 2006.

[5] E. Della Valle, D. Cerizza, I. Celino, J. Estublier, G. Vega, M. Kerrigan,J. Ram´ırez B. Villazon, P. Guarrera, G. Zhao, and G. Monteleone, "SEEMP: an semantic interoperability infrastructure for e-government services in the employment sector", *Lecture Notes in Computer Science*, Vol. 5419, 2007, pp. 220-234.

[6] Local Development Coordination Actions enabled by Semantic Technology (LD-CAST) Project. Available: http://www.ldcastproject.com

[7] "UML 2.0 Superstructure Specification", Object Management Group, 2005. Available: http://www.omg.org

[8] "Applying 4+1 view architecture with UML2", *FCGSS White Paper*, 2007

[9] J. Sommerville, P. Sawyer, "Viewpoints: principles, problems and a practical approach to requirements engineering", *Annals of Software Engineering* 3, 1997, pp. 101–130.

[10] Philippe Kruchten, "Architectural Blueprints—The "4+1" View Model of Software Architecture", *IEEE Software* 12 (6) November, 1995, pp. 42-50.

[11] R.E. Filman, Elrad T., Clarke S., Aksit M., "Aspect-Oriented Software Development", *Addison-Wesley Professional*, 2004.

[12] Based on a figure from A.D. Ionita, M. Florea, L Jelea, "Correspondence between Multiple Views in a SOA Trans-National Business System", *Lecture Notes in Engineering and Computer Science, Proceedings of the World Congress on Engineering 2009* Vol I, WCE 2009, July 1 - 3, 2009, London, U.K., pp. 493-498

[13] A. D. Ionita, A. Catapano, S. Giuroiu, and M. Florea, "Service oriented system for business cooperation". In *Proceedings of ICSE Int.*

Workshop on Systems Development in SOA Environments SDSOA '08, Leipzig, Germany, May 11 - 11, ACM, New York, 2008, pp. 13-18.
[14] Simple Object Access Protocol (SOAP) Version 1.2, Available: http://www.w3.org/TR/soap12-part1/
[15] JBoss Enterprise Middleware, Available: http://www.jboss.com
[16] S. Palkovits, D. Orensanz, D. Karagiannis, "Process modelling in Egovernment – Living process modelling within a public organisation", *IADIS International e-Society*, 2004, pp. 3-10.
[17] OASIS. "Web Services Business Process Execution Language for Web Services" Version 2.0, 2007. Available: http://docs.oasis -open.org/wsbpel/2.0/ wsbpel-v2.0.pdf
[18] F. D'Antonio, M. Missikoff, F. Taglino, "Formalizing the OPAL eBusiness ontology design patterns with OWL. Business episodes and workflow integration: A use case in LD-CAST", *Third International Conference on Interoperability for Enterprise Applications and Softwar*e, I-ESA 2007, pp. 345-356.
[19] M. Erdos, S. Cantor, "Shibboleth Architecture", Draft version 5. Available: http://shibboleth.internet2.edu
[20] A. Catapano, A. D'Atri, V. Hrgovcic, A.D. Ionita, and K. Tarabanis, "LD-CAST: Local Development Cooperation Actions Enabled by Semantic Technology", In *Proceedings of Eastern European eGov Days Conference*, EEGOV, Prague, Czech Republic, April 23-25, 2008, pp. 22-27.
[21] J. Sommerville, *Software Engineering*, 8th Edition, Addison-Wesley, 2006
[22] *The European e-Business Report 2008*, 6th Synthesis Report of the Sectoral e-Business Watch, Available: http://www.ebusiness-watch.org/
[23] Service Oriented Architectures for All (SOA4ALL) Project, Available: http://www.soa4all.eu
[24] Business objective driven REliable and Intelligent grids for real busiNess (BREIN) Project, Available: http://www.eu-brein.com
[25] H. M. Frutos, I. Kotsiopoulos, "BREIN: Business Objective Driven Reliable and Intelligent Grids for Real Business", *IBIS*, Issue 3 (1), 2009, pp. 39-42
[26] L. Kamara, B. Neville, J. Pitt, D. Ramirez-Cano, R. Chiriacescu, L. Dobrica, M. Florea and A. Szoke, "Regulatory compliance and alternative dispute resolution in e-societies", *IADIS International e-Society*, Algarve, Portugal. 9-12 April, 2008
[27] NESSI Open Framework - Reference Architecture (NEXOF-RA) Project, Available: http://www.nexof-ra.eu/
[28] D. Peltz, "Web services orchestration. A review of emerging technologies, tools and standards". *Hewlett Packard, Co*. January 2003.
[29] B. Stadlhofer, and P. Salhofer, "SeGoF: semantic e-government forms" In *Proceedings of the 2008 international Conference on Digital Government Research,* Montreal, Canada, May 18 - 21, 2008, ACM International Conference Proceeding Series, vol. 289. Digital Government Society of North America, 2008, pp. 427-428.
[30] J.M. Cantera, M. Reyes, J. Hierro, N. Tsouroulas, White Paper on Service Front-Ends in the Future Internet of Services and Next Generation SOA, Version 0.5, July 21st 2008
[31] Fast and Advanced Storyboard Tools (FAST) Project, Available at: http://fast.morfeo-project.eu/
[32] T. Herborn, A. Mondorf, B. Mougouie, M. A. Wimmer, Semantic interoperability in the BRITE project: Ontologies as a tool for collaboration, cooperation and knowledge management, In R. Meersman, Z. Tari (eds): On the move to meaningful Internet systems, *Lecture Notes in Computer Science,* Monterrey, Mexico, Vol. 5331, Springer, 2008, pp. 475-483.
[33] N. Stojanovic, G. Mentzas, D. Apostolou, Semantic –enabled Agile Knowledge-based e-government, in Proc. of *AAAI Spring Symposium. The Semantic Web meets eGovernment*, Stanford University, California, USA, March 27-29, 2006
[34] A. Mocan, E. Cimpian, M. Kerrigan, "Formal model for ontology mapping creation". In "The Semantic Web - ISWC 2006", *Lecture Notes in Computer Scienc*e, Springer, Volume 4273, 2006, pp. 459-472
[35] V. Peristeras, and K. Tarabanis, "Reengineering the public administration modus operandi through the use of reference domain models and Semantic Web Service technologies", In *Proc. of AAAI Spring Symposium on The Semantic Web meets eGovernment* SWEG, Stanford University, California, USA Mar. 27-29, AAAI Press, 2006, pp. 56-63.
[36] J. Estublier, S. Sanlaville, "Extensible process support environments for web services orchestration", *International Journal of Web Services Practices* (IJWSP), Vol. 1, No. 1-2, 2005, pp. 30-39.
[37] St. Roser, F. Lautenbacher, and B. Bauer, „Generation of workflow code from DSMs", *Proceedings of the 7th OOPSLA Workshop on Domain-Specific Modeling*, Montreal, Canada, October 2007.