Support for Forming Temporal Business Alliances as Networked Enterprises

Marián Mach, Peter Bednár
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
{marian.mach, peter.bednar}@tuke.sk

Karol Furdík
InterSoft, a.s.
Floriánska 19, 040 01 Košice, Slovakia
karol.furdik@intersoft.sk

Abstract. *Currently, business organisations face a problem how to cooperate on a case to case basis. Participation in temporary flexible business networks can provide a competitive advantage in the global market. The aim of the FP7 ICT EU project SPIKE is to design and implement a system that enables to support (in a technical way) a collaboration of business partners in a temporary business alliance involving the technical set-up of the alliance, running, and final closing-down of the alliance. The focus of the presented paper is on a designed architecture enabling different types of collaborations among business partners.*

Keywords. Business alliances, networked enterprises, semantic modelling, identity management, service-oriented architecture

1 Introduction

In digital economy, companies can compete not only regarding direct production costs but they can be successful with innovative business models, tight collaboration and division of services between sub-suppliers, partners, and subsidiaries. The key to success is flexible and fast reaction to market opportunities – recent years have shown the rise of business networks which can be realised by dynamic formation of organisational alliances.

Short-term business alliances have two main distinguishing characteristics: virtual partnership of organisations aiming at sharing partners' business services without restrictions on organisational size or structure, and ad-hoc process-oriented collaboration on the level of individual employees or working teams between partner organisations. Traditional software environments are not flexible enough to support these requirements in an ad-hoc way.

Development of a platform for inter-enterprise interoperability and business collaboration is the main technological and research objective of the FP7 ICT project *Secure Process-oriented Integrative Service Infrastructure for Networked Enterprises* (SPIKE, [9]). In particular, our aim within this project is to design and implement a system for enterprises of all sizes to be used for realising competitive advantage via forming business alliances. Advantages include the division of labour, specialisation to core competencies and collaboration for optimising processes of the value chain. The designed solution is based on four pillars:

- *Networked processes* by means of direct collaboration between core processes of involved organisations;
- *Business bus*, i.e. the alliancing based on high level of standardisation;
- *Electronic services* in a sense of cooperation employing standard externalised services;
- *Service integration* using "infomediaries" within business networking.

The system is expected to support an easy and fast setup of short-term business alliances, allowing dynamic addition and removal of alliance members. The solution encompasses a semantically enriched service-oriented infrastructure [3], including a service bus for message and process control, together with semantic filtering and transformation of messages. At user interface level, a collaborative process portal approach enables to capture users' working context to transfer it seamlessly to services according to a specified workflow.

The paper is structured as follows. Section 2 provides an overview of the scope and context of the proposed system. Section 3 presents an overall system architecture while attention is paid to different views

and perspectives of the architecture description presented within separate subsections. Section 4 focuses on various technologies employed for the system development. In section 5 we summarise the work done so far and discuss areas of future work.

2 Context and system boundaries

The scope of the system is determined by the envisioned functionality of the system as a whole, i.e. a technical support of a collaboration of business partners in a temporary business alliance. The system will support three phases of an alliance life-cycle (setting-up, running, and closing-down) considering the following three different forms or levels of collaboration:

- *Collaborative processes* that enable to produce physical or intangible artefacts and are modelled by means of complex collaboration patterns. The process consists of particular steps — activities representing contributions of the alliance partners to the collaborative output. These activities form a workflow with defined conditions, relationships, and an ordering of the activities.
- *Sharing services*, where the alliance partners can offer their services in the scope of a given business process. The offered services can be retrieved, negotiated, contracted, and finally used by the alliance members according to the conditions specified by the service contract.
- *Identity federation*, enabling and mediating the access of an alliance partner to the resources of other partners. It allows individuals (employees of an alliance partner) to get access to a network of a collaborative partner using the same credentials as they use in their home company.

The collaboration levels can be used independently one from another, but can also be combined together. For example, an alliance may offer a service composed of several steps that require participation of one or more partners. These steps can be glued together into a collaborative process — in this way, the collaborative effort of some alliance partners can be offered to the other partners as a service to be shared. In addition, consumption of a service may require the identity distribution and the control of accessing the resources of the participants.

The *system context* is given by those external entities (actors) that are expected to communicate with the system. The context diagram, depicted on Fig. 1, presents the highest level view of the system boundaries and its adjacent external entities. Human roles as *Administrator*, responsible for maintenance and day-to-day operation of the system, and *Service/process designer*, able to create internal composed services and collaborative processes, were defined for the alliance set-up and management. The *Service provider*, as an owner of a service included into the collaboration, is responsible for registering, contracting, and de-registering services.

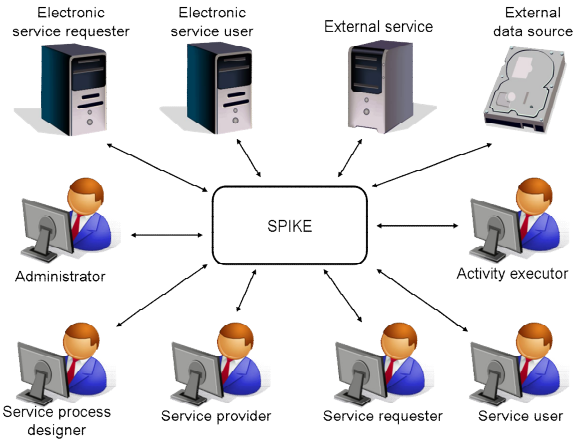

Figure 1. Context of the SPIKE system

External service is a role of a software actor that provides a specific functionality that can be accessed and consumed using a protocol that interfaces the service. *Activity executor* role represents a manual version of a service and is used when a specific functionality is not provided by an application but by a human actor only. *External data source* role stands for an information repository enabling to store and retrieve the external data (e.g. from database, information system, or document repository) using a specific interface.

Service users, both human and software actors, are service consumers that are able to obtain output values produced by a contracted service after submitting an input required by the service and triggering the service. Finally, *Service requesters* represent a prospective alliance partner (a human actor or a software agent) and are responsible for allocating relevant services and contracting them.

3 Overall system architecture

The methodology of Rozanski and Woods [8] was adopted for the architecture design by identifying the viewpoints, perspectives, and stakeholders of the SPIKE system. User partners of the project provided a description of required functionality [12], which was subsequently used as a background for specification of system views and perspectives as well as a platform for the validation of the system design. Some of the main views and perspectives are presented in the following subsections.

3.1 Functional view

The highest level functional system architecture, schematically depicted in Fig. 2, consists of four main functional subsystems:

- The *SPIKE System Core* (SSC) is a back-end that provides functions for processing all the system data;
- The *SPIKE Portal Instance* (SPI) is a graphical user interface and acts as a front-end to the SSC;
- The *SPIKE administration, monitoring and reporting* (SAMR) subsystem is a toolkit for system maintenance and day-to-day operation;
- The *SPIKE Service Bus* (SSB) enables internal communication between SSC, SPI and SAMR as well as communication with external entities.

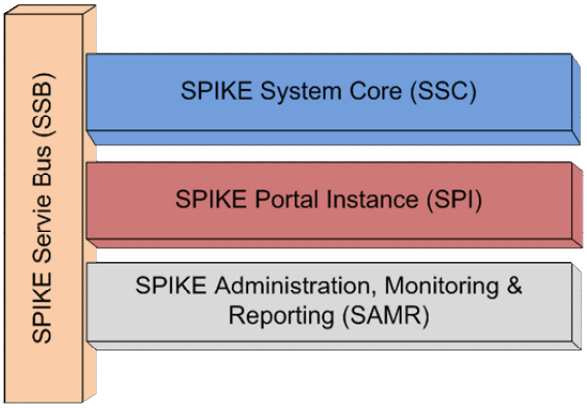

Figure 2. Functional architecture overview

Each of these main subsystems is further divided into several *managers* — system components providing autonomous and elementary functionality. The design of managers was accomplished according to the methodology of [1], namely trying to balance the coupling vs. cohesion and sufficiency vs. completeness metrics. We have preferred functional cohesion – packing functionally related parts into one element. The subsystems and particular managers, as they were specified for the SPIKE system, will be described in more details in the following subsections.

3.1.1 SPIKE Service Bus

The SSB subsystem is a central communication channel that handles messaging and data exchange between the individual parts of the SPIKE infrastructure. It consists of two managers, as depicted in Fig. 3.

The *Interface Manager* is the only component employed by the other managers to interact with external services. It provides basic capabilities for service usage, i.e. for connecting services and transmitting service requests and responses. As an extension of the Interface Manager, the *Communication Manager* provides more comprehensive and powerful transformation, enrichment and message routing capabilities. It consumes the functionality of some other managers. The Security Manager is employed for access control, authentication, and authorization, while the data mediation and semantic transformation of the service messages and notifications is performed by calling the Semantic Manager.

3.1.2 SPIKE System Core

The SSC subsystem provides core functionality as data storage, security, as well as maintenance of processes, workflows, and services. The internal structure of the SSC managers is presented in Fig. 3.

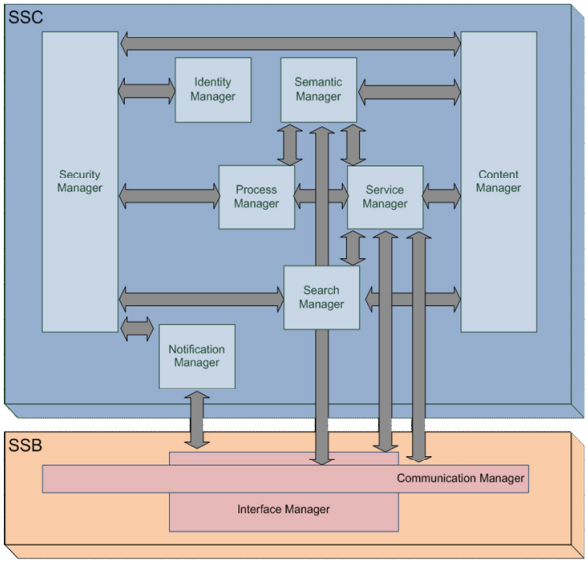

Figure 3. Managers of the SSB and SSC subsystems

The *Content Manager* is a very central component of the SSC. It is responsible for storage, retrieval and update of all the data presented in and brokered through the SPIKE infrastructure, namely the ontologies, service registrations and descriptions, user sessions, and metadata of identity management.

The security throughout all the processes handled via platform is ensured by the *Security Manager*, which provides functions of authentication, authorization, auditing, and ciphering. The authentication mode of the security is strongly related to the *Identity Manager*, which is responsible for handling individual users' identities (as credentials or user profiles) and making them available to all the partners in an alliance. The *Notification Manager*, as a central hub for notifications generated and consumed within the platform, provides alerting, triggering, messaging, and distribution of notifications among the alliance partners, system users and administrators.

The *Process Manager* is responsible for handling and executing processes expressed as workflows of activities and defined for an alliance. To be executed, the workflows need to be standardized and represented in the BPEL formalization [4]. The Process Manager provides supporting functionality for the workflow execution as customization of abstract services incorporated in the process,

deployment, initialization, and running of invoked executable workflows. The component also provides an execution environment for creating and carrying out human tasks (i.e. workflow activities performed by human actors), as well as for their management.

The *Service Manager* provides discovery and execution capabilities for the services integrated into a workflow. Internally, the module contains a web service engine that enables to access the external services via a web service interface. It also provides functions for analysis of the services registered in the platform, namely an estimation of selected Quality-of-Service characteristics as availability, performance, reliability, and capacity [5]. The service discovery is provided in cooperation with the *Search Manager*, which supports all facets of searching in the SPIKE platform. Besides the discovery of service instances, it includes the metadata-based and full-text search, as well as the semantic matching and filtering.

The *Semantic Manager* handles all the functionality involved in dealing with semantic information, namely the semantic search, matching, mediation, mapping, and reasoning over the semantically described data. Semantic metadata descriptions for business processes, services, and artefacts serving as input and/or output parameters of the services are maintained and provided by the Semantic Manager by means of metadata mapping.

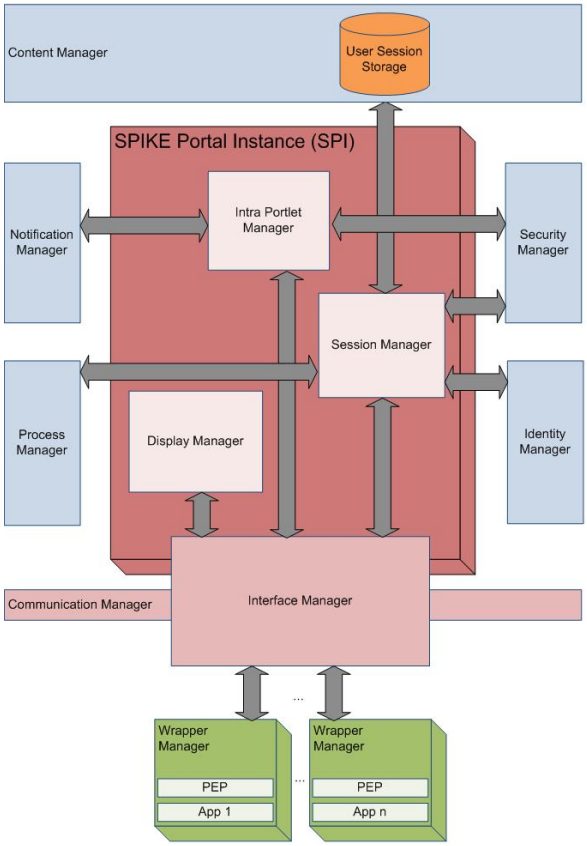

Figure 4. Architecture of the SPI subsystem

3.1.3 SPIKE Portal Instance

The SPI subsystem, schematically presented in Fig. 4, can be characterized as an end users' front-end to the system, aggregating information from the various sources available to the user via the SPIKE platform. Technically, the SPI combines multiple web applications, so-called portlets, into one single portal webpage.

The *Session Manager* is the central component within the SPI architecture. It is responsible for handling and keeping track of the whole context of a user session. The manager enables to store and retrieve workflow status information and all the user session-related data, using the storage repository provided by the SSC Content Manager.

The *Intra Portlet Manager* provides notification-handling facilities and distributes event information from external sources (i.e. other user sessions) to all destinations within one portal instance. As the manager collects events from all the sources within one session, it also works vice versa, delivering event information also to the SSC Notification Manager so it can inform other parties not involved in the current session but involved in a global workflow instead.

The *Display Manager* allows visualization of individual services connected to the platform via the SSB as well as visualization of internal data within the SPIKE. These internal data include, for example, the service execution information, process-related information, events received from an active user's session or from other sources within the platform, i.e. other alliance partners or services.

The *Wrapper Manager* is responsible for integration of existing legacy applications that cannot be represented via web service technologies. In other words, this manager is a direct link to an external target application intended for use within a collaboration, which does not offer a service-oriented interface. It is designed as a generic component that allows flexible adaptation on various types of legacy applications (typically, GUI-based as Windows/X11, or Web applications based on PHP, J2EE, etc.). One particular Wrapper Manager instance will then be specifically tailored to at least one particular application class, such as Windows native applications, providing wrapping, encapsulation, data exchange, security and identity federation.

3.1.4 SPIKE Administration, Monitoring, and Reporting

System management and reporting facilities are carried out within the SAMR subsystem, whose internal structure is depicted in Fig. 5.

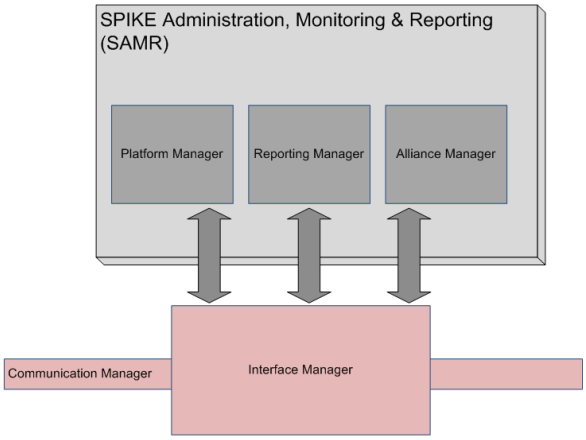

Figure 5. Managers of the SAMR subsystem, connected to the SSB

The *Platform Manager* provides means for administration and monitoring of the whole SPIKE platform. It shows the status of all the infrastructure components, the underlying databases and application servers and, additionally, acts as an interface to starting and stopping them. Therefore, the Platform Manager can be considered as a console fundamental to the installation and configuration of the SPIKE platform at rollout-time instead of being considered an integral part of the infrastructure itself.

The *Alliance Manager* is responsible for offering administration facilities of the business alliances handled via the SPIKE platform opposed to administration facilities of the whole platform, which are handled by the Platform Manager. Another distinction to the Platform Manager is that the Alliance Manager is integrated into the SPIKE's user portal, making use of the SPI's application integration capabilities. Consequently, the Alliance Manager is designed as a portlet and is included into an alliance administrator's portal instance by means of the Interface Manager.

Finally, the *Reporting Manager* supports functionality of generating various output reports over the services, processes, and other SPIKE data, together with a distribution of the reports across the platform. The reports are considered in the HTML and PDF output format, persistently stored in the repository maintained by the Content Manager. Distribution of reports to the respective addressees is performed using the message routing facilities of the SSB managers, while the Display Manager of the SPI provides visualization of reports on a proper place and in a proper format.

3.2 Information view

The information view of the architecture describes the way in which the system stores, manipulates, manages, and distributes information [8]. Identification of the information resources in their mutual relationships, information flows and data distribution was accomplished by analysing the user requirements [12] and resulted in a design of the structure depicted in Fig. 6.

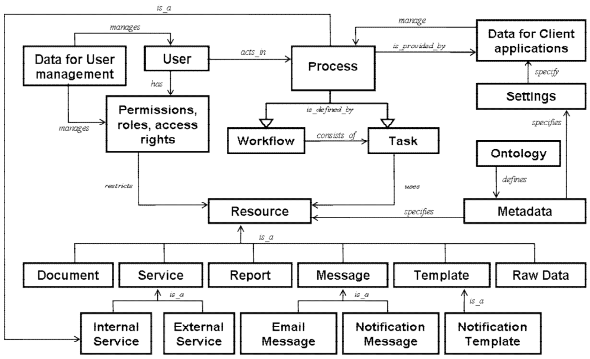

Figure 6. Data elements and structural correlations

Five basic information resources together with internal data elements were designed for the platform, as described in the following paragraphs.

- The *Business process model data* represent a model of business processes defined and operated within the platform. The *Process*, *Workflow*, and *Task* elements were specified as basic logical data components for the process modelling;
- The *Physical information resources* represent a storage space for the objects that will be manipulated by actors in a business process. The physical objects as documents (i.e. multimedia files of various formats), web services, email messages, reports created from a predefined templates, etc., are considered to be the data elements. In Fig. 6, the *Resource* data element is presented as a generic parent, i.e. an abstract "collaboration object" that defines a common set of properties inherited by the child data elements – particular resource types as *Document*, *Service*, *Report*, etc.
- The *Semantic metadata* stores and provides both the schema as well as the instantiated data that specify and semantically describe the elements of other information resources. The data elements of *Ontology* and *Metadata* elements are stored as a set of ontologies in the WSML format.
- The *Security and authentication data* stores and manages the data about users, their identities, profiles, preferences, roles, permissions, and access rights as far as needed by the security architecture.
- The *System data* information resource contains all the information needed for configuration of the SPIKE's client-side tools, global settings, system and environment properties. The data elements in this information resource are structured and will be stored in property files or optionally can also be stored as metadata in the system ontology.

3.3 Conceptual perspective

The system is expected to be able to employ semantic information to support its functionality. The employment of semantics is envisioned in the following areas:

- Provision of a *common vocabulary of terms*, i.e. a shared knowledge model, as a conceptual representation of a particular domain. Conformity with standards, assuming that some parts of the knowledge model will be reused from already existing and available ontology resources, is expected.
- *Interoperability and conceptual integration of services and/or resources* originating from heterogeneous platforms, locations, access rights, and usage policies on a semantic basis. Metadata description of services, annotated by concepts of a shared knowledge model. Optionally, the discovery, composition, and orchestration of the services by means of their conceptual descriptions, formal logic expressions / constraints and context of the service within a business process, can also be supported.
- *Mediation* of properties with the same (or similar, related) meaning, but expressed by different forms (e.g. "Date of birth" vs. "birth date"). Unification of several different ways of conceptualisation, ontology merging.
- *Reasoning*, inference of new (i.e. a-priori unknown) facts from an existing knowledge model. Inheritance of concept attributes, complex multiple relations, rules and constraints.
- *Retrieval* of the information or data, based on its semantic metadata description. Access to the data according to their meaning and semantically expressed (i.e. conceptualised) context.

In addition, for the data maintained by system managers, the semantic layer can provide a "mirror" data storage using semantic metadata description. In this way, the data can be wrapped and semantically related to the rest of the data stored in the system.

However, the semantics is not considered as a core but additional feature of system components, which evolutionary extends their functionality. In this case, the system components are only loosely dependent on semantic descriptions and can proceed even if there is a lack of semantic metadata caused by incomplete annotation of resources (semantic graceful degradation).

The core of the semantic layer is a knowledge model containing *conceptual schema* and *metadata instances*. The schema is represented by a structure of ontologies and provides a basic conceptual framework. The instances represent semantic metadata consumed by functional building blocks.

The knowledge model (both its levels) is maintained by the *Semantic Manager*. This manager provides all the functionality necessary for dealing with semantic information as maintenance of the model, mediation, matching, reasoning and semantic retrieval over the conceptual schema and metadata as well as other functionality supporting semantic discovery, composition and resolving of services, participating in collaborative processes.

3.3.1 Ontology structure

The following ontologies have been proposed for modelling semantic information within the system:

- *Core ontology*, containing basic common concepts of upper generalisation level, shared among the application cases. It also defines a conceptual framework for collaboration environment of organisations, including concepts as Collaboration object, Organisation, Address, Person, Actor, etc.
- *Resource ontology* for description of physical information resources – the data elements as Document, Report, Message, and Template.
- *Service ontology*, describing the information resources of Service type. Internal and external services are defined, together with such properties as location, access means, goals, inputs, outputs, preconditions, and effects.
- *Domain ontologies* for the definition and description of domain-specific concepts, relations, transition rules, and business logic constraints for particular domain of interest, e.g. for a single application case. Domain ontologies will also define a structure of the data entries as strings, date, time, or currency values, etc., localised for various language versions.
- *System ontology* for the description of global settings, system and environment properties, as well as for the configuration data of client-side tools.
- *Business process ontology* for semantic extension of business processes towards mapping BPEL elements to concepts of a common and standardised ontology. This ontology semantically extends and enriches the data specified and stored in the Process Manager component of the system.
- *User ontology* for description of information on users, user profiles and preferences, roles, identities, access rights, etc. The user ontology can be considered as a semantic extension of the data on users specified and stored in the Security Manager.

4 Technology employed for system development

From the implementation point of view, the SPIKE system is divided into three packages, which can be deployed and used on the stand-alone machines.

The first package, corresponding to the SAMR subsystem, roughly covers the *design tools* needed to create all the configuration artefacts required to set-up alliances. It is expected that the creation of these resources will require some expertise, namely when dealing with the semantic and process modelling as well as during the implementation of web services.

The SAMR subsystem is envisioned to be implemented as the set of plug-ins integrated in the Eclipse IDE platform (http://www.eclipse.org). It consists of tools for BPMN modelling, including facilities of export to BPEL processes [11], the tools for designing the interfaces for human tasks, implemented using the XForms, the toolkit for the design of ontologies and semantic mediator rules, and tools for semantic annotation of services using the SA-WSDL annotations.

The tools implemented in the SAMR will be based on the existing tools such as BPMN Modeller, Visual XForms Designer, and WSMO Studio. The output of these design tools will be provided as an alliance assembly bundle, i.e. an archive file containing all the configuration artefacts, including ontologies required to establish new alliances, and a deployment descriptor for automatic deployment of the bundle to the SSB.

The second package stands for a user interface to the SPIKE platform and covers mainly the functionality of the SPI subsystem. It will be implemented as a set of portlets integrated in the standard JSR 168 Portlet container. The portal interfaces are designed for the Administrators (cf. section 2), who will be responsible for deployment, start, and monitoring of the alliances between various business partners. The second type of the user, who will use the portal interface, is the Activity executor. This user role will perform the human tasks included in the processes of the alliance interactions. The portlet component for human tasks will be based on the user interface provided in Intalio Tempo framework (http://www.intalio.org) for the BPEL4People processes.

The last package covers functionality of the SSB and SSC subsystems. It will be implemented as the JSR 208 Java Business Integration (JBI) compliant *enterprise service bus*. We have selected the Apache ServiceMix ESB to be suitable for the JBI container implementation. Almost all the SSC components will be implemented as the JBI components, i.e. as the service engines such a BPEL execution engine, which is the part of the Process Manager, or as the JBI binding components for various protocols or external applications, which will be part of the Interface Manager. The standard Normalized Message Router (NMR) of the JBI component will be used as the main communication channel for delivering messages between the SSB components.

The most innovative part of the SSB consists of components that will extend functionality of the JBI environment by a semantic-based dynamic binding and semantic data mediation. In the JBI environment, all executed BPEL processes need to be defined as abstract, i.e. without the binding of the process partner links to the specific services. During the execution of the BPEL process, all partner links have to be resolved to the specific services, which will provide requested functionality. The dynamic semantic-based resolving of the services will provide a matching of semantic descriptions specified for the interfaces of process partner links and of provided specific services. Semantic description of the interfaces will be specified by means of the SA-WSDL annotations, which will map various WSDL elements to the semantic models (i.e. ontologies). The SPIKE ontologies will be specified using the WSML family of ontology languages. The semantic data mediation will also be implemented using the combination of SA-WSDL/WSML specifications. In the first step of the mediation, normalized messages sent from a service requester through NMR will be transformed (i.e. lifted) into the semantic instances. Semantic inference will then be used to infer output instances, which will be transformed (i.e. lowered) back to the mediated messages sent to/from the service provider. Lifting and lowering transformations will be specified using the SA-WSDL annotations, while the rules for mediation inference will be expressed by the WSML rule languages.

The implementation of the semantic functionality in the Semantic Manager will be based on the software packages developed for the WSMO Lite semantic framework [10]. The main component of the Semantic Manager is the in-memory object model of the WSML language elements implemented in the wsml4j package. The wsml4j package is connected to the underlying inference engine (i.e. a reasoner) using the wsml2reasoner API. Connection to the ontology repository is accomplished by means of the Ontology Representation and Data Integration (ORDI) framework. We have selected IRIS rule inference engine as the reasoner for the WSML rule languages.

The security infrastructure, considered as very important and crucial for the business collaboration system, was designed for SPIKE in terms of attribute/role management, authentication, workflow and service access control, and auditing functionality. Rather than an independent component, the Security Manager should be seen as a horizontal layer that influences most of the platform components and managers. A hybrid mechanism of SASL [7] and GSS-API [6] was proposed for authentication, integrated with the WS-Security protocol for secure web services. The authorization will be handled by

the PERMIS infrastructure [2], providing facilities to manage users' privileges and authorization policies. Auditing mechanism enables to trace user actions such as accessing resources of involved companies or using provided services. The non-repudiation approach to the auditing will be employed to provide compulsory certified tracing by means of confirmation services and timestamps, with possibility to add the digital signatures in future.

5 Conclusions and future work

The architecture of the SPIKE system, presented in the paper, integrates business process modelling capabilities and principles of the Semantic Web to enable creation and maintenance of temporal business alliances. This approach is supported by employing the technologies such as enterprise service bus, semantic business process modelling, portlet-based user interface, and advanced security infrastructure. The efforts should result in the development and provision of a generic collaboration environment that can successfully serve as a platform for inter-enterprise interoperability.

At the time of writing the paper (April 2009), the architecture design and specification of functional subsystems is already finished and implementation of the platform components is ongoing. According to our plan, first release of the implemented system should be ready in September 2009. After that, the platform will be tested within the first trial on three pilot applications in business organisations from Austria and Finland. More information on the project can be found at [9].

6 Acknowledgments

The SPIKE project is co-funded by the European Commission within the contract No. FP7-ICT-217098. The work presented in the paper was also partly supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/4074/07 Project "Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge".

References

[1] Booch G, Maksimchuk R A, Engle M W, Young B J, Conallen J, Houston K A: **Object-Oriented Analysis and Design with Applications**, Third Edition, Addison Wesley Professional, 2007.

[2] Chadwick D W, Otenko A, Ball E: **Role-Based Access Control with 470 X.509 Attribute Certificates**, IEEE Internet Computing, vol. 7, No 2, March-April 2003.

[3] Hepp M et al: **Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management**, Proc. of the IEEE ICEBE 2005, October 18-20, Beijing, China, 2005, pp. 535-540.

[4] Jordan D, Evdemon J et al: **Web Services Business Process Execution Language 2.0**, OASIS Standard, 11 April 2007, available at `http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html`, Accessed: 24th April 2009.

[5] KangChan L et al: **QoS for Web Services: Requirements and Possible Approaches**, W3C Working Group Note 25 November 2003, available at `http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/`, Accessed: 24th April 2009.

[6] Linn J: **Generic Security Service Application Program Interface**, Version 2, Update 1. RSA Laboratories, January 2000.

[7] Melnikov A, Zeilenga, K: **Simple Authentication and Security Layer (SASL)**, Isode Limited, OpenLDAP Foundation, June 2006.

[8] Rozanski N, Woods E: **Software Systems Architecture. Working with Stakeholders Using Viewpoints and Perspectives**, Addison Wesley, 2005.

[9] **SPIKE project**, `http://www.spike-project.eu`, Accessed: 24th April 2009.

[10] Vitvar T, Kopecky J, Fensel D: **WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web**. WSMO Deliverable D11, Ver.0.2. DERI, 2008.

[11] White S A: **Using BPMN to Model a BPEL Process**. IBM Corp., 2005.

[12] Wiesbeck S et al: **D2.2 User Requirements Analysis & Development/Test Recommendations**, SPIKE project FP7-ICT-217098, Public Deliverable, 2008.