An Agent-based Bridge between Business Process and Business Rules

Agostino Poggi
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma
Viale G.P.Usberti 181/A, 43100 Parma, Italy
+39 0521 905728
Agostino.poggi@unipr.it

Paola Turci
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma
Viale G.P.Usberti 181/A, 43100 Parma, Italy
+39 0521 905708
Paola.turci@unipr.it

ABSTRACT

Industry has been and still is, more than ever, interested in executing business functions that span multiple applications. This demands high-levels of interoperability and a more flexible and adaptive business process management. In the attempt to delineate an effective solution, some researchers have envisaged as strategic the integration of the service-oriented technology with business process management, semantic and Web technologies. Others have turned their attention towards agent technology, integrated with the aforementioned technologies, as an interesting means for implementing business systems.

In these years, our research work has been mainly devoted to improving the integration of the JADE agents with the most interesting knowledge and internet-oriented technologies. In this paper we present our latest effort consisting of the integration of workflow and rule technologies in multi-agent systems. The approach is based on the use of rule engines as part of service implementation and the use of business process engines for service orchestration. Agents play a mediation role and their coordination properties are essential for the realization of an increasingly flexible and effective business process management.

Keywords

Multi-agent systems, workflow, rule engine, business process management.

1. INTRODUCTION

The most successful software infrastructure technologies of the last ten years have explicitly targeted the integration of heterogeneous, distributed software applications. All the popular frameworks for software infrastructure, from RPC to Web Services, have the common trait of being a specification for component boundaries and interaction patterns. Such technologies share a common promise: to ease the integration of heterogeneous software systems and to facilitate the maintainability of the overall system. To reach this goal, they all inject conceptual integrity into heterogeneous software systems through a model that coherently encompasses the whole system.

In the last years, a new paradigm has emerged as the most appropriate response to this need: the service-oriented architecture (SOA). To be exact, there is no a single recognized definition of SOA. However, a baseline of concepts and principles and a strategic vision have emerged and collectively characterize the service-oriented design paradigm as an approach to defining integration architectures based on the concept of service. Recently, several analysts have believed that enterprises can achieve their critical goal to support business agility and transformation by applying SOA principles to the Business Process Management in a synergistic fashion.

The widespread adoption in the industrial sector of the service-orientation design paradigm within the business process architecture, as evidenced by technology and market research companies' surveys, is having a twofold influence on the evolution of the agent technology.

On the one hand, several researchers belonging to the agent community are convinced that this technical area is a natural environment in which the agent technology features can be leveraged to obtain significant advantages. Due to the introduction of the social level of system description, the multi-agent systems approach looks particularly well suited to addressing complex, heterogeneous distributed systems. A major aim of multi-agent systems is to enable software integration on a deeper level, namely shifting the integration process from syntactic interoperability to semantic interoperability, providing conceptual integrity to heterogeneous, distributed software systems.

On the other hand, one of the requirements for the success of multi-agent systems is that they have to guarantee easy integration with other widely used industrial technologies.

The information and the research activities in this area are, however, quite fragmented; the subject of business transformation and optimization turns out to be vast and enormously complex and more work needs to be done in order to realize real flexible, adaptive, intelligent business applications.

Trying to deeply analyzing the motivations of the SOA success we can draw some conclusions. The main reason of using a service-oriented architecture is clearly to increase enterprise agility and minimize the impact of inevitable changes. This is achieved through separating the parts of the business process that often change from the ones that are fairly stable. In a typical SOA scenario, such objective is fulfilled by the definition of services, generally Web services, which are dynamically composed in order to rapidly adapt to the new business needs. Besides the way in which services are composed, another component of the business process that frequently changes is the set of the business rules applied. The use of rules in business modelling is becoming more and more important, especially in applications requiring dynamic changes of behaviour. A number of rule languages and tools has been proposed to the software engineering community and the need to effectively integrate business process with business rule is increasingly felt.

The work presented in this paper fits in this context and can be seen as a continuation of our activities oriented towards the improvement of JADE framework. In particular, in this paper we address the implementation of an agent-based bridge between business process and business rule, trying to maintain the necessary separation of concerns of the two systems.

Section 2 gives a brief survey of the literature in the area of business process management, focusing mainly on the JADE framework. It provides an overview of the scenario in which our work is situated. The third section describes the integration of business rules and workflow technology in the JADE agent development framework. Section 4 describes a simple but realistic application which shows how the powerful synergism between agent and business management technologies could be very promising. The paper ends by drawing some conclusions around the results of the work done.

2. BACKGROUND AND RELATED WORK

There is evidence from several research studies [10],[14] that agents represent one of the most suitable technologies which can be used to meet the performance needs for the conception and realization of real flexible, adaptive intelligent business applications. Indeed, agents provide both an appropriate level of abstraction in modelling complex and distributed applications and a natural merging of object orientation and knowledge-based technologies that can facilitate the incorporation of reasoning, learning and high-level dialogue capabilities to realize intelligent and adaptive applications. In particular the current interest in using agents as a valuable support to advanced enterprise applications is rising mostly because different works have shown how agent technology can be leveraged if used together with the cornerstone technologies of mainstream computing [1][2][8][12].

In the following, the current status of such key technologies is analyzed from the viewpoint of integration with agent-technology and particularly with the JADE framework.

2.2. Agents and Web Services Composition

In general, the modus operandi to carry out Web service compositions is similar to that concerning the definition of workflows, such that existing techniques for workflow pattern generation, composition, and management can be partially reused for this purpose [7]. As a matter of fact, the mainstream approach to Web service composition is characterized by a manually definition and a centralized execution of Web services based workflows. The manual definition is reasonable since in an enterprise context, characterized by business partners binding by a legal contract, unlike in open service environment, there is only a very limited need for automated service composition planning,

Therefore, restricting the area of investigation and focusing mainly on the standardization efforts related to service composition based on workflow patterns for the business process management, a prominent standard that provides for explicit process modelling and execution, based on Web services, is the business process execution language for Web services (WS-BPEL).

The WS-BPEL [20] specification defines an XML-based language for the formal description of a business process based on Web services orchestration. It is an open standard approved as an OASIS standard. A WS-BPEL workflow is a structured XML document composed of three main parts: (i) the definition of the process attributes; (ii) the definition of the execution context, and (iii) the activities to be executed. This language is highly expressive and offers some flexibility by allowing workflows to refer to abstract service interfaces rather than concrete instances. This means that services can, in principle, be selected dynamically at run-time, depending on current service availability. Due to industry's increased interest on business process management and the wide acceptance of WS-BPEL as the language to use in the workflow definition, several vendors are producing software tools for workflow design, specification and enactment.

WS-BPEL is particular significant if used in combination with another W3C candidate recommendation: i.e. the Web service choreography description language [21]. In fact, the two specifications are complementary: WS-BPEL is about orchestration, it is based on a central coordinator that is responsible for invoking and combining the single sub-activities, while WS-CDL is about the modelling and execution of cross-organizational business processes, specifying the common observable behaviour of all participants engaged in business collaboration.

The main drawback of this standard specification is that it requires service instances to adhere exactly to the syntactic interfaces specified by the workflow designers. The decisions about composition take place during design-time when the architecture and the design of the software system are planned, in other words WS-BPEL does not provide support for dynamically discovering or exploiting (in case of unpredictable events) of new Web services, offering a semantically equivalent service but with different syntactic interfaces.. This may be suitable on condition that business partners and service components do not or only rarely change, but in a dynamic environment characterized by a continuous evolution where companies need a business process that is capable of ongoing modifications, this often represents an undesirable limit.

Relying, indeed, exclusively on XML one can reach only a syntactic interoperability. Expressing message content in XML does not make possible semantic "understanding" of the message contents and as a consequence programmers have to reach explicit agreement on both the format of the messages exchanged and the way Web services interact.

On the one hand the syntactic description is essential since it provides information about the structure of input and output messages of a service and how to invoke the service itself, but on the other hand semantics is needed to describe what a service really does. Semantic Web technologies promise to address this by offering more formal service discovery and composition techniques, thus allowing applications to use previously unseen and heterogeneous services at run-time with a limited human intervention.

2.3. Web Services, Workflow and Rules in JADE

Several works proposed the integration of JADE with Web services and workflow technologies (see, for example, [11],[13],[9][17][18]. However, the JADE official solution for such integration is based on the JADE WSIG add-on and on an extension of JADE, called WADE, that supports the use of workflows in the realization of multi-agent applications.

JADE supports the invocation of agent services as Web services and the capability to realize applications as composition of agents and Web services through a JADE add-on called WSIG - Web Service Integration Gateway [5]. WSIG is able to automatically expose agent services as Web services and to convert SOAP invocations into ACL requests. More in details, JADE agents publish their services in the Directory Facilitator of a JADE multi-agent platform. Each registered service is described via a data structure called Service-Description. This structure specifies, among others data, the ontologies that must be used to access the published service and that define the actions that the agent is able to perform. WSIG listens to registrations with the Directory Facilitator and, for each registered agent service, it automatically exposes a Web Service described by a WSDL description whose operations correspond to the actions supported by the registering agent. If properly configured, WSIG is also able to publish the exposed Web Service in a UDDI registry in order to simplify integration of a JADE-based system within a SOA environment. When an agent needs to invoke a Web service it directly creates the SOAP message and sends it to the provider, e.g. exploited AXIS2 API. In this case no particular support is provided by WSIG. To date, WSIG supports only simple WSDL description of Web services, without taking into account emerging technologies related to the semantic Web.

WADE (Workflow and Agent Development Environment) is a software platform, built on top of JADE, for the development of distributed applications based on the agent-oriented paradigm [4]. WADE adds to JADE the support to the workflow execution and a few mechanisms to manage the complexity of the distribution, in terms of administration and fault tolerance. WADE adds some additional components to a JADE application:

- A boot daemon on each host on which the application is deployed with the duty of activating the JADE containers of the application on the current host.
- A configuration agent on the main container of the application. This agent is responsible for interacting with the boot daemons and controlling the application life cycle.
- A controller agent for each container in the platform and they are responsible for supervising activities in the local container and for all the fault tolerance mechanisms provided by WADE.
- A set of workflow engine agents able to execute workflows.

In particular, the workflow engine agents embed a workflow engine able to executed workflows encoded by an extended version of XPDL (Workflow Management Coalition) that adds the possibility of associating the direct execution of pieces of Java code with the activities of a workflow. In fact, the main challenge in WADE is to bring the workflow approach from the business process level to the level of system internal logics, i.e., its main goal is not to support the high level orchestration of services provided by different systems, but the implementation of the internal behaviour of each single system. Moreover, the execution of a workflow can be shared by a set of workflow engine agents because each workflow engine agent can delegate the execution of some subflows to some other agents.

Finally, JADE provides the integration with the JESS rule engine which is probably the most known Java rule engine implementing the Rete algorithm. This integration is realized through the so-called JessBhaviour that allows the encapsulation of a JESS rule engine inside a JADE agent and has the duty of storing and retrieving information in/from the rule engine. The main limits of this solution are: i) the rule engine is completely hidden to the other agents of the system and there is not any support for the cooperation among different rule-based agents (i.e., agents encapsulating a JESS rule engine) and ii) JESS is a commercial software and so we have additional costs if we plan to realize commercial applications by using JADE together with JESS.

3. DROOLS RULEFLOW INTEGRATION

In order to cope with the limits of the current JADE support for rule engines, we realized a software library, called D4J (Drools4JADE) [12], that integrates JADE agents with the Drools rule engine [3]. Drools is a well known, freeware implementation of the so-called Rete-OO algorithm. Apart of its open-source availability, one of the main advantages of Drools is exactly the fact that it is not just a literal implementation of the Rete algorithm, but rather an adaptation for the object-oriented world. This greatly eases the burden of integrating the rule engine and the application rules with the existing external objects. In Drools, asserted facts are simple Java objects, that can be modified through their public methods and properties. Where Jess requires hundreds of lines of code, for example to simply access an ACL message mapped into a Java object, Drools rules can obtain the same result in a dozen of easy-reading code lines.

D4J guarantees both the advantages of full rule-based agents, i.e., agent whose behaviour and/or knowledge is expressed by means of rules, and the advantages of rule-enhanced agents, i.e., agents whose behaviour is not normally expressed by means of rules, but that use a rule engine as additional component to perform specific reasoning, learning or knowledge acquisition tasks. In facts, in D4J, the Drools rule-engine is integrated into an agent as a JADE behaviour, but it also provides an API for interacting with it through ACL messages allowing both remote storing and retrieval of knowledge and the cooperation among different rule-based agents. Moreover, this API allows rules mobility, i.e., a rule-based agent can move a rule to another rule-based agent.

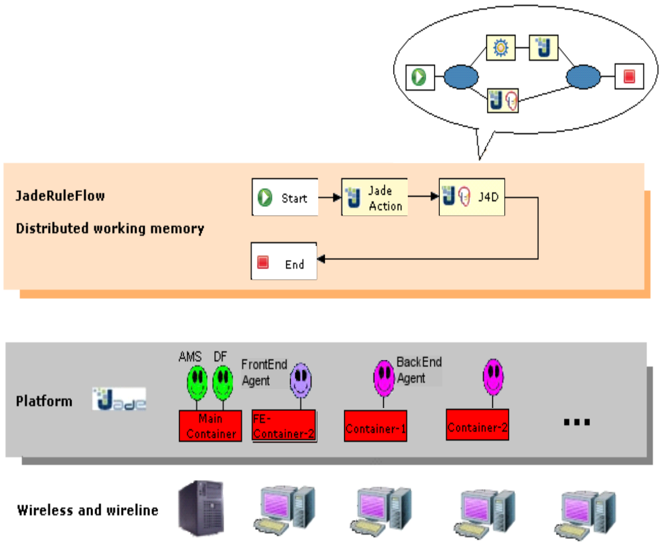


Figure 1. JadeRuleFlow architecture.

In these last months, D4J has been extended to take advantage of workflow extension of Drools (Drools Ruleflow) that allows the realization of rule based workflows. In fact, Ruleflow allows the specification of the order in which rule sets should be evaluated by using a flow chart and so allows the definition of which rule sets should be evaluated in sequence or in parallel and the specification of the conditions under which rule sets should be evaluated. In particular, our work was dedicated to realized an extension of JADE, we called JadeRuleFlow, for the distributed execution of rule based workflows. Figure 1 shows a graphical description of JadeRuleFlow architecture.

JadeRuleFlow is based on two types of agent: the frontend agent and the backend agent. A frontend agent is the interface between the

Drools software for the management of RuleFlows and the JADE agents used for their execution. In particular, this agent, manages the execution of the workflow delegating tasks and waiting for their completion and the updating of the facts associated with the rules involved in the workflow. A backend agent has the duty of processing a set of facts received by the frontend agent and returning the updated set of facts to such an agent.

The work that must be done to realize a JadeRuleFlow application consists in the development of the behaviours that the backend agents will use for processing Drools facts. In fact, the creation of the frontend agents and the backend agents involved in the execution of a rule workflow is automatically done by a software tool called Jade for Drools J4D. This tool extract from the RuleFlow the information about the behaviours to be executed and about the data (facts) that are needed for their execution, then it creates the frontend agent and initialized it with the data necessary for the creation and synchronization of the backend agents.

4. AN APPLICATION EXAMPLE

In today's competitive market, companies strive to increase the quality of products or services provided. The process of complaint handling is related to products and services within an organization, including planning, design, operation, maintenance and improvement, and it is part of the overall quality management system of an organization. The process aims to enhance the organization's ability to improve its products, process and customer services. A coarse-grain description of the process is shown in Figure 2. It represents the workflow schema of the entire process, a quite stable procedure suitable for the implementation by means of a business process engine. Each "macro-activity" represented in it can be further detailed. As far as the non-conformity handling activity is concerned, in Figure 3. a possible workflow is shown. Any work that is not conducted in accordance to the specified work-process is a non-conformity and should function as a signal for improvement or correction. The non-conformity management is a very delicate task, heavily prone to modification and thus the use of business rules is preferable in this case.

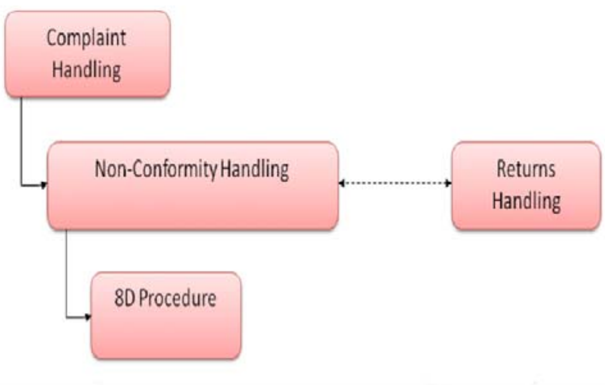


Figure 2. The activities involved in the complaint management process.

Assuming to adopt an agent-based approach for implementing a workflow system, the architecture of the multi-agent system would be composed of a heterogeneous society of agents, where different members play different roles, are responsible for specific work items and have a different internal complexity. In such a heterogeneous society, hierarchical collaboration between reasoning capable agents is achieved mainly through goal delegation.

As far as the complaint handling is concerned, the most relevant types of agents, which compose the society, are: workflow manager agent, complaint handling manager agent, non-conformity manager agent, 8D procedure manager agent, returns handling manager agent and personal assistant agent. In order to achieve their goals these agents are able to perform their tasks in cooperation with each other and most of them can dynamically change their behaviour based on business rules.

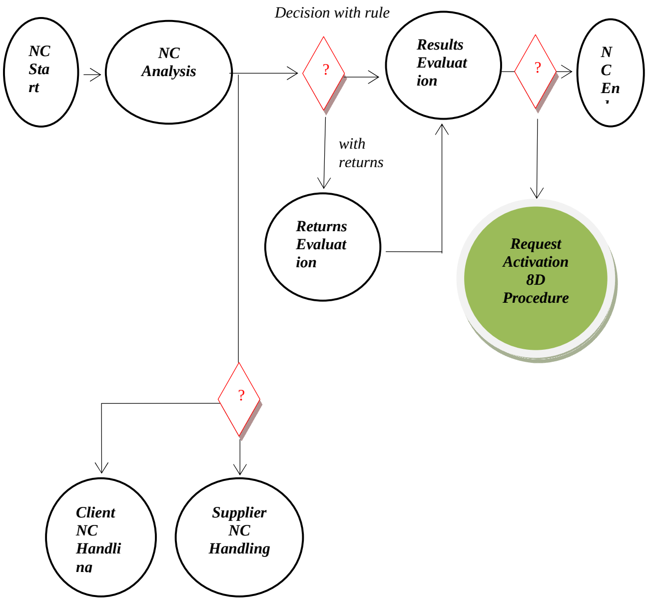


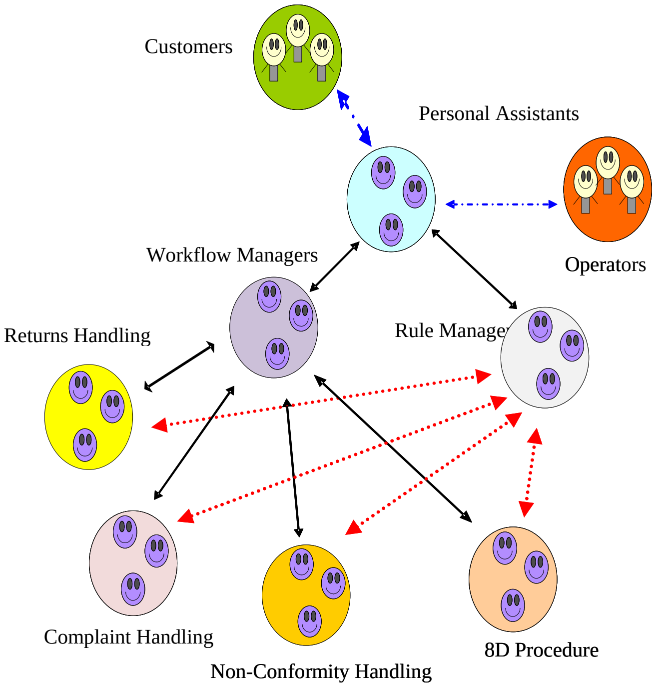
Figure 3. The non-conformity handling workflow

Workflow manager agents have the goal of executing and monitoring WS-BPEL workflows. Rule manager agents are in charge of supporting the operators in the maintenance of the business rules. The request of a modification, insertion or deletion of a rule will propagate from the rule manager to the manager agents involved. The whole business process, described by the WS-BPEL workflow, is broken into pieces or activities which are embedded into the remaining manager agents. For instance, the Non-conformity Handling Manager Agent is responsible for providing the service which handles observed and reported non-conformity. While such manager agents are responsible for carrying out specific activities, made available as Web services, the workflow manager agents are responsible for the orchestration of such activities.

Finally, personal assistant agents, associated with each user active in the system, are responsible for the interaction between the user and the other parts of the system. In particular, they are responsible for helping operators in the maintenance of business rules and customers in lodging complaints.

When a customer asks its personal assistant to lodge a complaint, the personal assistant will delegate the handling of such complaint to a workflow manager. This agent will be responsible for initiating and coordinating the workflow enactment, relying on the Web services provided by the other agents of the system. The enactment is clearly a problematic phase. When a workflow is going to be executed, a Web service could be no more available due to a failure of a resource or other unpredictable problems. In this case the workflow manager will create a new contract phase with all the manager agents that are able to satisfy the task and will carry out the replacement of the failed service with a new one.

The architecture of the multi-agent system is represented in Figure 4.



igure 4. Interactions between the different types of agents in a complaint management systems.

5. CONCLUSIONS

In our view, business process and business rules are complementary technologies. Business processes define a set of activities that need to be performed in order to achieve certain goals. Business rules provide a higher level of flexibility and configurability; an ideal means to facilitate the adaptation to rapidly changing business environments. In this paper we have addressed the issue concerned the integration of business process and business rules. We have referred to an interesting pattern used in SOA implementation, i.e. the exploitation of rule engines as part of service implementation and the use of business process engines for service orchestration. As a matter of fact, service orchestration typically deals with long-running coordination and asynchronous invocation of external activities/services. Business process engines have turned out to be the most appropriate paradigm for this purpose. Business rules, on the other hand, are useful as a supplement to the workflow schema, more stable, for reaction to certain events, more prone to changes, and thus they are more suitable for adaptation aiming to satisfy business needs.

The research in distributed artificial intelligence has been addressing for several years the problem of designing and building coordinated and collaborative intelligent multi-agent systems. This interesting and advanced work can be fruitfully exploited in the area of business process management if agent technology is appropriately engineered and integrated with the other key technologies. Firmly convinced of these advantages, we have delineated a possible approach trying to show how the powerful synergism between these technologies could be very promising.

6. REFERENCES

[1] Buhler P.A., Vidal, J.M. (2005). Towards Adaptive Workflow Enactment Using Multiagent Systems. Information Technology and Management, 6(1):61-87

[2] Bergenti, F., Poggi, A., Tomaiuolo, M. & Turci, P. (2005). An Ontology Support for Semantic Aware Agents. In by J. G. Carbonell and J. Siekmann (Eds.), Agent-Oriented Information Systems III, Vol 3529, (pp. 140-153). Berlin, Germany:Springer.

[3] Drools rule engine Web site (2009). Retrieved June 15, 2009, from http://www.jboss.org/drools/.

[4] Giovanni Caire, Danilo Gotta, Massimo Banzi: WADE: a software platform to develop mission critical applications exploiting agents and workflows. AAMAS (Industry Track) 2008: 29-36.

[5] Greenwood D. & Calisti M. (2004). Engineering Web Service-Agent Integration. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pages 1918–1925. The Hague, Netherlands.

[6] Domingue J. & Fensel D. (2008) Toward a Service Web: Integrating the Semantic Web and Service Orientation, IEEE Intelligent Systems January/February, 2008.

[7] L. Henoque, M. Kleiner: Composition - Combining Web Service Functionality in Composite Orchestrations. Chapter 9 in [104], 2007. R. Studer, S. Grimm, A. Abecker (eds.): Semantic Web Services. Concepts Technologies, and Applications. Springer, 2007.

[8] Kishore R., Zhang, H., & Ramesh, R. (2006). Enterprise integration using the agent paradigm : foundations of multi-agent-based integrative business information systems, Decision Support Systems, 42(1), 48–78.

[9] Liu, S., Küngas, P., & Matskin, M. (2006). Agent-Based Web Service Composition with JADE and JXTA. In Proc. of the 2006 International Conference on Semantic Web and Web Services, (pp. 110-116), Las Vegas, NV.

[10] Luck M., McBurney, P., Shehory, O., & Willmott, S. (2005). Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing), AgentLink. Retrieved March 31, 2009, from http://calcium.dcs.kcl.ac.uk/778/1/al3roadmap.pdf.

[11] Martinez, E. & Lespérance, Y. (2004). IG-JADE-PKSlib: An Agent-Based Framework for Advanced Web Service Composition and Provisioning. In Proc. of the AAMAS 2004 Workshop on Web-services and Agent-based Engineering, (pp. 2-10), New York, NY.

[12] Negri A., Poggi A., Tomaiuolo M., Turci P. (2006). Agents for e-Business Applications, In AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. (pp. 907-914). Hakodate, Japan. ACM Press

[13] Nguyen X. T. (2005). Demonstration of WS2JADE. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pages 135–136. Utrecht, The Netherlands

[14] Papazoglou1 M., Traverso P., Dustdar S., Leymann F. (2006). Service-Oriented Computing Research Roadmap.

[15] Savarimuthu B. T. R., Purvis M., Purvis M. & Cranefield S. (2005). Integrating Web Services with Agent Based Workflow Management System (WfMS). In WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence. (pp. 471 – 474). Washington, DC. IEEE Computer Society.

[16] Shafiq M. O., Ali A., Ahmad H. F., Suguri H. (2005). AgentWeb Gateway - a Middleware for Dynamic Integration of Multi Agent System and Web Services Framework. In Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative

Enterprise, pages 267–270, Washington, DC. IEEE Computer Society.

[17] Soto, E.L., (2006). Agent Communication Using Web Services, a New FIPA Message Transport Service for JADE. In P. Petta J.P. Müller, M. Klusch M. Georgeff (Eds.), Multiagent System Technologies, Lecture Notes in Computer Science, Vol 4687 (pp. 73-84). Berlin, Germany: Springer.

[18] Trappey, C.V., Trappey, A.J.,C., Huang, C., & Kud, C.C. (2009). The design of a JADE-based autonomous workflow management system for collaborative SoC design. Expert Systems with Applications, 36(2), 2659-2669.

[19] Workflow Management Coalition. Workflow Process Definition Interface - XML Process Definition Language (XPDL). Version 1.0,, 2002. Available from http://wfmc.org/standards/docs/TC-1025 10 xpdl 102502.pdf.

[20] WS-BPEL. (2007). Web Services Business Process Execution Language OASIS Standard. Retrieved March 31, 2009, from http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html.

[21] WS-CDL. (2005). Web Services Choreography Description Language W3C Candidate Recommendation. Retrieved June 135, 2009, from http://www.w3.org/TR/ws-cdl-10/.