Web Agent's Enclaves – A New Opportunity for the Semantic Web Services

Violeta Damjanović
Postal Savings Bank
27.marta 71, Belgrade, Serbia and Montenegro
vdamjanovic@posted.co.yu

Dragan Gašević, Vladan Devedžić, Dragan Djurić
FON – School of Business Administration, University of Belgrade
POB 52, Jove Ilića 154, Belgrade, Serbia and Montenegro
gasevic@yahoo.com, devedzic@galeb.etf.bg.ac.yu, dragandj@mail.ru

Abstract

The paper presents an idea about joint use some of new technologies such as the Semantic Web, Web services and Web agents together with an already existing concept. It is a concept of enclaves that has been used on Workload Manager (WLM) that provides connection optimization on IBM's operating system, z/OS. In accordance to the future Web's vision as a semantics-driven construct of Web services and ontologies, we have looked for an answer about methodology to discover complex Web services, as well as to discover complex processes that will be able to appear between ontologies into the Web. As an efficient answer, we have detected the Web agent's society with characterized organization in the form of enclaves. We have introduced the Web agent enclaves as open, flexible, and dynamic Web agent's societies with the function of permanent observation and analysis of complex processes, which arise between ontologies on the Web.

1. Introduction

The idea about the way that is possible to use in managing within the future Web environments comes from z/OS. Because the space of the future Web is becoming unexplored in the sense of the further use of traditional ways for its searching and organizing of knowledge, we have looked for an answer from z/OS, as well as from new Web technologies. By aggregation of some existing idea and new technologies, we have found an interesting approach to organize the future Web functionality with the aim of achieving the basic task of the future Web – quick searching through semantically marked and correct information from ontology knowledge domains.

After briefly introduction, some of basic definitions and notions from new Web technologies areas, such as the Semantic Web, Web services and Web agent systems, as well as from introduced mechanism from z/OS, and its adaptation to the current Web environment, are explained in section 2. Section 3 briefly explains joint role of the Semantic Web, Web agents and Web services from the perspective of mechanism for creation an ontology process enclave. Section 4 gives a simple example of creating an ontology process enclave in order to show a new Web agents society. Section 5 presents our current development in order to enable building XML-supported knowledge bases that will be used in inference engines of process enclaves. This solution is a part of the GOOD-OLD-AI research group (http://goodoldai.org.yu) efforts to develop a platform for building Semantic Web enabled intelligent systems.

2. Web technology aggregation

In this section, we will explain the way on which it is possible to connect three various Web technologies, as well as the mechanism with which the joint use of all technologies can contribute to the new Web functionality.

2.1. The Semantic Web & Ontologies

The future generation Web, called the Semantic Web is a vision of new way of Web knowledge organization, which originates from the form of decentralized vocabularies [16] – ontologies. Apart from ontologies, in the background of the future Semantic Web intelligence there are knowledge bases, inference engines, and also standards that make possible reasoning with the marked concepts on the Web. But, when we talk about the Semantic Web, we firstly think of ontologies. Ontologies can be considered as knowledge bases about specific domain, and reality of further spreading of Web, which implies appearance of a great number of domain ontologies in a semantically marked-up world of the Web. Also, ontologies can be described in terms of different aspects of an architecture they refer to solving different kinds of collaboration or integration problems [15]. As a result, we have following types of ontologies such as shown in Table 1.

Table 1. Types of Web ontologies

Type of ontology	Description
A role-based (or generic) ontology	• Defines the terminology and concepts relevant for a particular end-user (person or consumer application). Generic ontologies provide super theories, like knowledge about *is-a* or *part-of* relation.
A process (or application) ontology	• Defines the inputs, outputs, constraints, relations, terms, and sequencing information relevant to a particular business process or set of processes. Usually, it contains all the necessary knowledge for modeling a particular domain.
A domain (or classic) ontology	• Defines the terminology and concepts relevant to a particular topic or area of interest (e.g., electronic, medical, mechanic…).
An interface ontology (Frame Ontology)	• Defines the structure and content restrictions (such as reserved words, units of measure requirements, other format-related restrictions) relevant for a particular interface (e.g., application programming interface (API), database, scripting language, content type).
Upper ontology	• Defines data interoperability, information search and retrieval, automated inferencing, and natural language processing.

2.2. The Web Services

The future Web doesn't consider only the use of the domain knowledge in the field of the Semantic Web technology, but the use of knowledge, which comes in different times, from the different sources, in different forms, which must be equally accessible and recognizable for the different Web applications. One of the central roles of ontologies is to establish further levels of interoperability, i.e. semantic interoperability between Web agents and applications on the emerging Semantic Web [18]. In other words, introducing of the Semantic Web technology in Web services functionality represents basic step towards establishing new Web generation. Such kind of the Web concept we can find under the definition of the Semantic Web enabled Web Services (SWWS) [9]. Generally, Web services are applications, which can be described, published, located and called anywhere on the Web [8]. At the same time, Web services have the ability of automatic location of other services [12]. The future plan about Web services is giving ability of automatic composing into more complex services needed in resolving the more complex requests. This ability of Web services will be realized via Web services choreography and Web services orchestration, as a kind of Web services management mechanisms. There is a big difference between Web services choreography and orchestration, which have explained in [7] [13]:

- Choreography represents a way for tracking the sequence of messages that may involve multiple parties and multiple sources. It is a way for public messages exchanges that occur between multiple Web services.
- Orchestration of Web services represents mechanism that describes the interaction between Web services based on messages. At the same time, orchestration considers business logic and order of execution of interaction activities of Web services, which will be shown in this paper as the scenario of Web agent's enclave use in the role of the Orchestration engine.

Apart from management mechanisms, some of basic functions of Web services given in [18] are shown in Table 2. Also with the use of the Semantic Web technology existing Web service functionalities can be broadened, as well as functionalities of the whole Web. These new features of Web services are suggested in [8] [9], and we give a short overview of this extension in Table 2.

As a main factors of Web services technology, which simultaneously represents Web services standards, we can mention following:

- Universal Description, Discovery and Integration (UDDI) – a standard for description, online registration, announcing and dynamic finding of announced Web services [1];
- Web Service Description Language (WSDL) – a language that describes the interface of a Web service, information about calling the service, and where to find it [17];
- Simple Object Access Protocol (SOAP) – a main technology for message exchange in Web service architecture [18].

Table 2. Web service's goals

Web service's goals	Description
Interoperability	• The fundamental goal of Web services is to surpass differs between the various environments that exist into the current Web.
Scalability and extensibility	• The Web services architecture must be scalable and extensible.
Security and reliability	• The Web Services architecture must be reliable and stable, and also must provide a secure environment for online processes.
Web-friendly	• The Web Services architecture must be consistent with the current and future evolution of the WWW.
Manageability	• The Web Services architecture must have definition the role of the manager [14].
Execution monitoring of Web services	• To address the problem of Web service monitoring, it can be used integrated methods of requirements analysis and software execution monitoring.
Automated composition of Web services	• Automatically generate a composition of Web services [10].
Recovery	• To recover data, whether it has been accidentally erased, or if data is missing due to a virus or malfunctioning software over an Internet connection.
Simulation	• Simulate the evolution of a Web service under different conditions [10].
Verification	• Automatically establish that the Web service upholds specified properties [10].
Validation	• Determination of the correctness of the Web services with respect to the user needs and requirements.

2.3. The Web agents

It is possible to do semantic marking of existing registered Web services, their descriptions, characteristics, possibilities, interfaces and effects [9] by the use of the Semantic Web technology. Thus, it is possible to prepare Web services for use by the Web agents. The role of Web agents in the new Web space will be changed compared to the present Web distributed environment. The main reason is different approach to the information on the future Web compared to the existing Web environment. Today, Web agents search the Web spaces via key words, not through their contents. In the future Web, Web agents will use semantically marked and understandable information represented on some language, which will be readable to them. Thus, we can explain the use of the Semantic Web technology for broadening of Web services possibilities via Web agent's functionality.

The role of the new Web agent societies in the future Web is indubitable. From this reason, we introduce a new kind of Web agent's society we can explain via the notion of enclaves. We can begin the story about the Web agent's enclaves from the definition of an enclave we have found in literature about z/OS environment: *An enclave is an anchor for a transaction that can be spread across multiple dispatchable units in multiple address spaces* [5]. The value of using an enclave to represent a transaction is that the resources used to process the transaction can be accounted within the transaction itself, rather than to the address space or spaces that the transaction runs in. In other word, an enclave consists of programs and resources for running threads (tasks) [3]. Each enclave can be one of the following types [5]:

- Independent enclave – represents a complete transaction. Also, the independent enclaves are used to represent new transactions that have not yet been associated with an address.
- A dependent enclave – represents the continuation of an existing address space transaction under a new set of dispatchable units.

On the contrary of enclaves from z/OS, concept of enclaves on the Web will be different. There are no more users who define performance criteria for an enclave, as a collection of performance goals and processing capacity rules. The whole process during the life cycle of enclave on the Web is event-driven. An event is semantically marked as a process within the Process Ontology, and we can say that it is a semantics-event-driven, or more simple semantics-driven. Because the semantic marking of processes on the Web implies the use of ontologies in the form of process ontologies that can serve the agent systems on the Web, we introduced the notion of ontology process enclaves that we will use in this work. Basic interest is focused towards discovering processes that arise between existing Web ontologies. Also, the moment of creating process enclaves as parts of Web agent's societies is important, and will be presented as follows.

3. Ontology process enclaves

We have introduced and defined the term of ontology process enclave as a Web multi-agent system that forms itself in dynamical and flexible way from existing Web agent systems with the purpose of constantly *control*, *analysis,* and *realization* of complex processes that arise between ontologies on the Web. The control and making recommendations about use of certain ontology in some time is also the mission of these Web societies, as well as use of certain processes as management mechanisms for further searching the spaces of Internet not only from ontology process enclaves but also from Web agents outside of these societies.

The relationship between the Semantic Web and Ontology area, Web services area, and Web agent area described in Section 2, is shown in Figure 1. Web agents from the heterogeneous Web environments become members of ontology process enclave according to the specific process, which is necessary to develop, control, and analyse. Determination which process on the Web is important to do by use of ontology process enclave is task of an enclave's own mechanism for reasoning about Web processes and their execution sequences. These processes are semantically marked in the Process Ontology that arises as a result of ontology process enclave's work during its life cycle. On the other and, all of Web services registered by the UDDI, can be semantically marked and enrolled in the Process Ontology. Web agents from ontology process enclave contact this Process Ontology not only from the reason to enroll some of the results, but also to define what is their next mission. Figure 1 shows the connection of the Semantic Web ontologies, Web services and Web agent systems, which will allow semantically exchange and synergy effects from near future of Web technologies.

As it has been shown in Figure 1, the Process Ontology represents initial mechanism from which the enclave's inference engine forms its membership, recognizes business logic of registered Web services, and inference about the sequence of Web service's interaction activities. Thus, the Process Ontology allows ontology process enclave functionality in the form of the Web service's Orchestration engine, as one of possible Web management mechanisms. For orchestration, the process is always controlled from the perspective of one of the business parties [13]. In the case we have shown in this paper, the control business parts on the Web are ontology process enclaves and their own inference engine. Figure 2 depicts the way how ontology process enclaves work, as well as the way of their inference features.

On the Web agent's area, there are many of heterogeneous and substantive Web agents, as well as different Web agent societies. In this paper, we have suggested a kind of Web agent's society in a form of ontology process enclave. An enclave is the basic unit of contents management (for each process on the Web there is an enclave, and resources can be share among all enclaves), as well as performance management (makes the best use of its resources and maintain the highest possible throughput and achieves the best possible Web system responsiveness by exploiting enclaves, monitoring enclaves, and managing enclaves).

Depending on complexity of Web services and processes, it's happened an exchange of complexity level on ontology process enclave, and than we can differ independent, and dependent enclaves. Figure 2 explains an independent, as well as a dependent enclave we can describe as follows:

- An independent enclave Enc-1: enclave's inference engine recognizes certain request from Web agent's area, and concludes about creation a new enclave, Enc-1. After receiving task request, and joining an ontology process enclave Enc-1, the task is executed. If the task is a simple and predefined by the enclave's inference engine it will be done, and thus, enclave Enc-1 will be finished.
- Dependent enclaves Enc-2, Enc-2', and so on: if enclave's inference engine concludes about necessity for creation enclave Enc-2, and during their life cycle some change happens discovered by enclave Enc-2, enclave's own inference engine can concludes about necessity for creation new sub-enclave Enc-2'. Depending on complexity of processes, the number of ontology process sub-enclaves can be multitude.

4. Ontology process enclave - an example

A simulation of creating an ontology process enclave is shown on JADE local platform, which provides start of numerous agents. Figure 3 shows a JADE agent's platform on which are started one main container, and several independents agent's containers.

JADE main container contains following agents [2]:

- Remote Monitoring Agent (RMA), agent that is implemented as Graphical User Interface (GUI) and serves for the remote management, monitoring and controlling of the status of agents, allowing to stop/restart agents;

- Directory Facilitator (DF), which allows to register/ deregister/ modify/ search for agents and services;
- Agent Management System (AMS), for automatic agent registration.
- Also, within main container we have started a Jess agent which basic role is to enable an inference mechanism about next actions that should be executed by ontology process enclave. In other words, the Jess agent represents an enclave's inference engine that works as an instrument for agent self-organization and adaptation to the Web environment. In this stage of developing ontology process enclaves as Web agent's societies, the Jess agent has a simple implementation, and it is strictly connected with the Jess rule file (with clp extension).

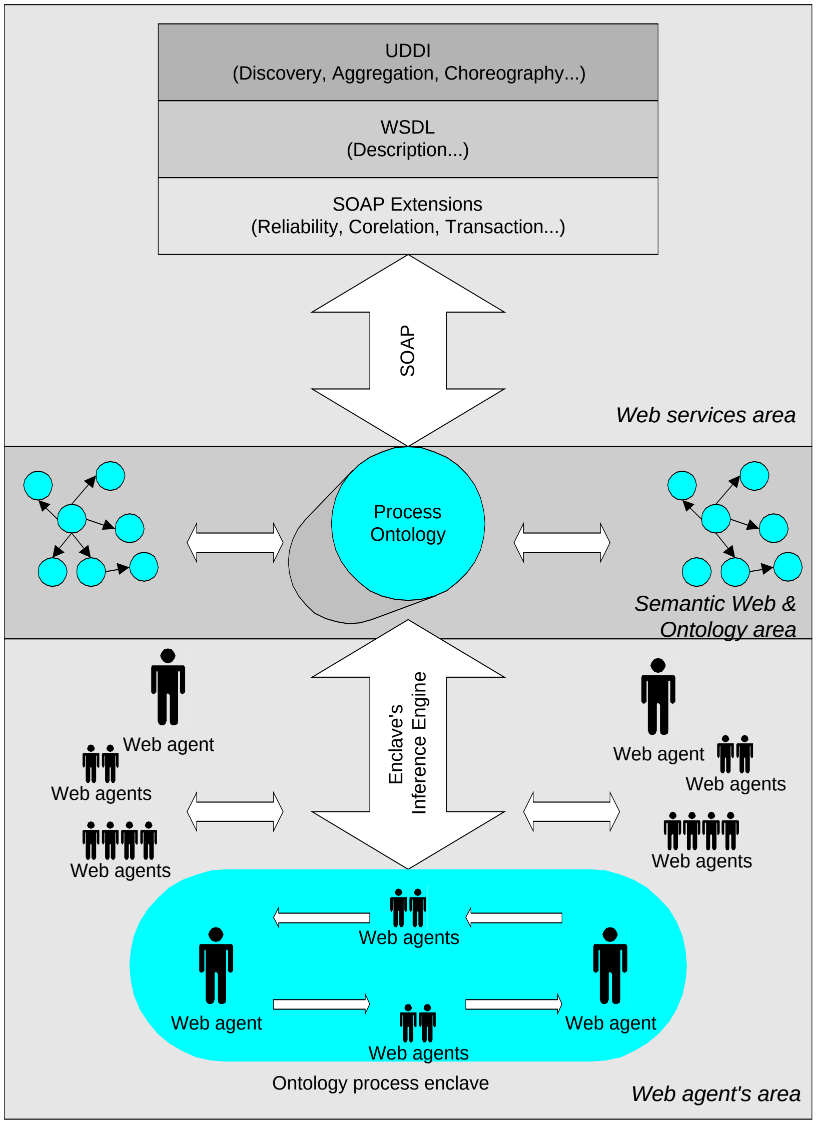

Figure 1. Relationship between the Web services, Semantic Web ontologies, and Ontology process enclave

On the same JADE platform several enclave's containers have started, from Container-1 to Container-5 shown in Figure 3. These containers contain agents that come from heterogeneous Web environment, and represent parts of ontology process enclaves. Also, these agents are process ontology-based what is implemented via Protégé-2000 ontology development editor, but it's outside of the scope of this work. Due to JADE possibilities to start up additional JADE agents such as Introspector agent, Dummy agent, and Sniffer agent, it is possible to compose Agent Communication Language (ACL) messages and send them to other agents, intercept ACL messages, and displays them graphically. Thus, it is possible to simulate the life cycle of ontology process enclaves.

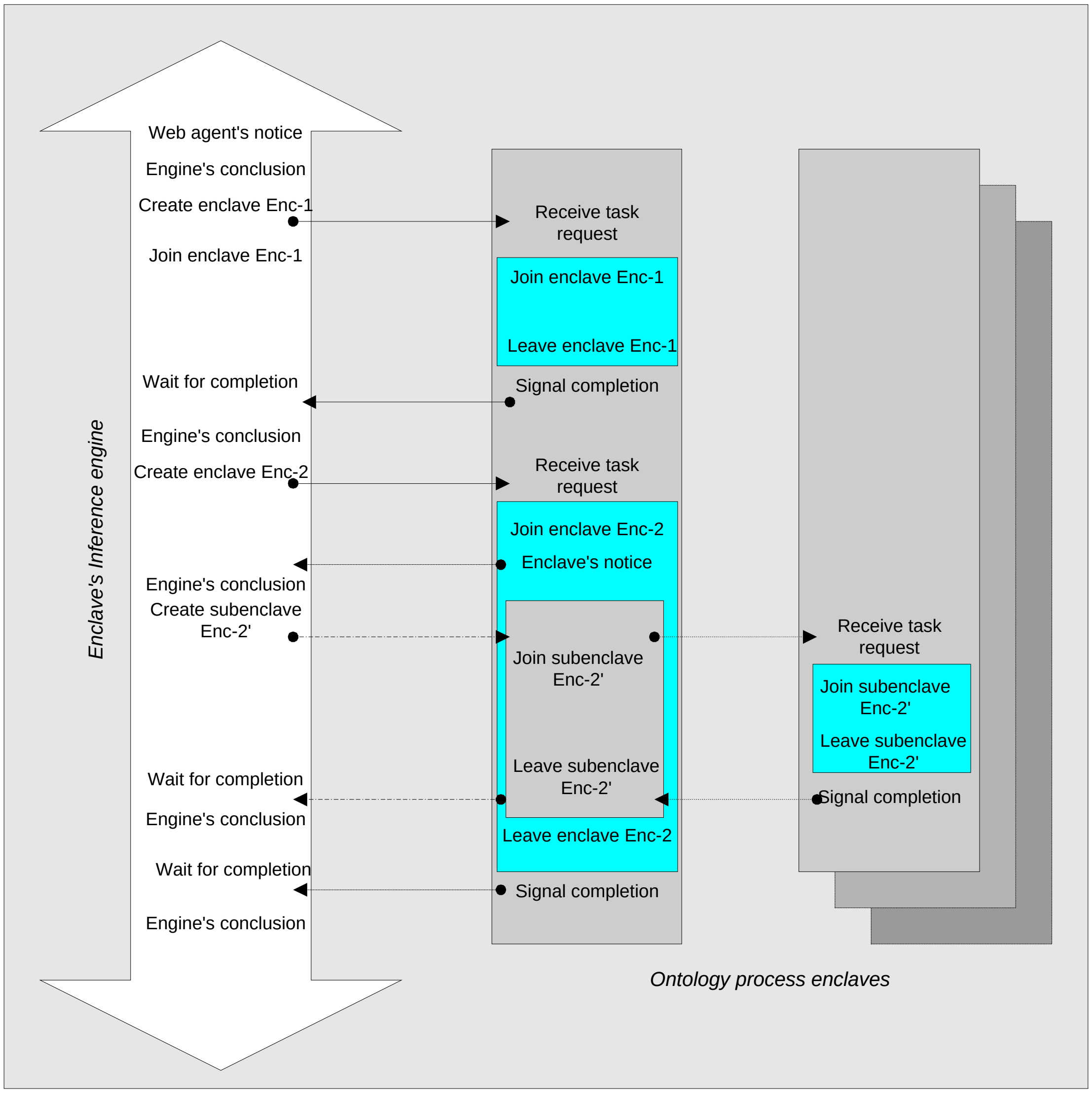

Figure 2. Realization of Orchestration engine via Ontology process enclaves

Tools needed to perform the implementation of the ontology process enclaves are shown in Figure 4. We can observe these tools in the context of their role on the Web agent's societies, so we have the following tool segments:

- Web agents tool segment – JADE;
- Semantic Web ontology tool segment – Protégé-2000 ;
- Inference engine tool segment - Jess;
- Tool segments for relationships setup – include the following plug-ins: the beanGenerator, the JadeJessProtege, and the JessTab plug-in.

By considering an inference engine tool segment, Jess and their appearance in the JADE tool as Jess agent, we can notice that there is a need for flexible use of Jess rule defined in the Jess (clp) rule file. That is the reason we have suggested a development a Jess agent in the GUI form and with possibility of flexible creation of new rules for ontology based research and inferences. Accordingly, in the next section we describe our ongoing work in developing a suitable GUI and its integration with well-established frameworks for developing intelligent systems.

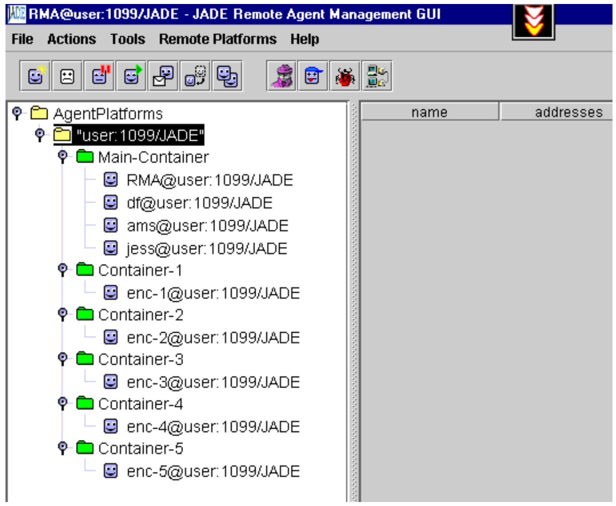

Figure 3. A simulation of ontology process enclaves on local JADE platform

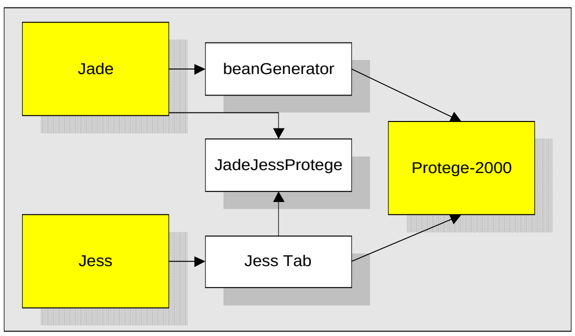

Figure 4. Relations between Web agents and Web ontology development tools

5. Extending enclave's functionality – our on-going work

We propose development of an appropriate GUI in order to create more flexible way for manipulating the Jess-based agent. Our efforts are directed to integrate Jess agent with our graphical tools for building knowledge bases.

The first approach to create a suitable GUI for the Jess knowledge bases is the JessGUI environment [6]. An important feature of JessGUI is its capability of saving knowledge bases in XML format (in addition to the original Jess format), thus making them potentially easy to interoperate with other knowledge bases on the Internet. However, the Jess interpreter in its basic distribution interprets only Jess/CLIPS code. Accordingly, we developed an XSLT that transforms the JessGUI's XML format into the Jess/CLIPS format, and this XSLT is performed in JessGUI when one wants to export a knowledge base into a Jess/CLIPS knowledge base. Using the same XSLT principle we implemented conversion of the JessGUI's XML format into RDF(S) and OWL. However, this tool still requires knowledge of the Jess syntax (e.g. for defining functions or THEN parts of rules).

The second approach is based on a more general framework for building intelligent systems – OBject Oriented Abstraction (OBOA) [4]. The OBOA is a multi-layered framework that incorporates a number of intelligent system techniques for knowledge representation. Following recommendations given in the OBOA we have implemented the JavaDON tool. JavaDON has capacity to build rule and frame-based intelligent systems. It supports saving knowledge bases in its own XML format as well as saving rules in the RuleML format (http://www.ruleml.org). In JavaDON we have implemented two inference mechanisms: forward and backward chaining.

The JavaDON tool and Jess agent are weakly coupled in the current development stage. In Figure 5 we depict their connections. JavaDON produces a knowledge base in the XML format. We developed an XSLT that transforms JavaDON's XML documents into the JessGUI's XML format. Finally, the JessGUI's XML format is transformed into the Jess/CLIPS format through the XSLT.

In the future we are planning to use a JavaDON based agent. That means, we will not use Lisp-like Jess format, but only XML-based knowledge bases. Also, we are developing JavaDON classes in the form of the API. This way, agents, which act as an inference mechanism in process enclaves, will be able to use the JavaDON's support for reasoning. In order to enable ontology-based communication between agents (e.g. process enclave) we will implement transformation of OWL or RDF(S) based ontologies into a Java class the JADE agents are able to consume. This approach has similarity with approach given in [11] where agents exchange UML-based ontologies and transform them into Java classes using XSLT mechanism.

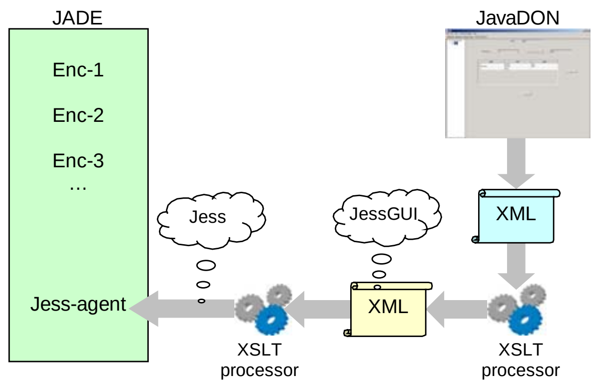

Figure 5. Integration of JavaDON and Web agents in process enclaves

6. Conclusions

In this paper we suggested the use of specific Web agent's society in the form of ontology process enclave. The main aim of the ontology process enclave's use will be to enable a class of Web management mechanisms for content management, as well as for performance management. Also, we have shown a possible use of ontology process enclaves as managing mechanism for realization Web service orchestration mechanism.

Our future work is directed towards research of new challenges that appear with the use of this mechanism. Also, we are planning to improve current version of the JavaDON tool that will be served to inference about the use of existing, registered processes, as well as about combining new processes with the aim for realization specific purpose within ontological process domains.

References

[1] Ankolekar, A. et al., "DAML-S: Semantic Markup for Web Services," In Proceedings of Proceedings International Semantic Web Conference, Sardinia, Italy. pp. 348-363.

[2] Bellifemine, F. et al., 2003. JADE Programmer's Guide. [Online]. Available: http://sharon.cselt.it/projects/jade/

[3] Comstock, S., 2000. z/OS, Language Environment, and UNIX: How They Work Together, 2002 [Online]. Available: http://www.trainersfriend.com/Papers/zos_unix.pdf.

[4] Devedžić, V., Radović, D., "A Framework for Building Intelligent Manufacturing Systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C - Applications and Reviews, vol. 29, no. 3, 1999, pp. 402-419.

[5] IBM, 2000. OS/390 V2R9.0 MVS Workload Management Services (GC28-1773-07). IBM Corporation, Inc. [electronic edition].Available: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IEA1W220/CCONTENTS?DT=20000119151125.

[6] Jovanović, J., Gašević, D., Devedžić, V., "A GUI for Jess," Expert Systems With Applications, Vol. 26, No.4, pp. 625-637.

[7] Kreger, H., "Fulfilling The Web Services Promise," Communications of The ACM, Vol. 46, No.6, 2003, pp 29-34.

[8] McIlraith, S. and Martin, D., "Bringing Semantics to Web Services," *IEEE Intelligent Systems*, Vol. 18, No. 1, 2003, pp. 90-93.

[9] McIlraith, S. et al., "Semantic Web Services," *IEEE Intelligent Systems*, Vol. 16, No. 2, 2001 pp. 46-53.

[10] Narayanan, S., McIlraith, S., "Simulation, Verification, and Automated Composition of Web Services," *In Proceedings of the 11th International World Wide Web Conference*, Honolulu, Hawaii, USA, 2002, 77-88.

[11] Pan, J., Cranefield S, Carter, D., "A Lightweight Ontology Repository," In Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-agent Systems, Melbourne, Victoria, Australia, 2003, pp. 632-638.

[12] Paolucci, M. et al, "Importing the Semantic Web in UDDI," *In Proceedings of Web Services, E-Business and Semantic Web Workshop*, Toronto, Canada, 2002, pp. 225-236.

[13] Peltz, C., 2003. Web Services Orchestration - a review of emerging technologies, tools, and standards. [Online]. Available: http://devresource.hp.com/drc/technical_white_papers/WSOrch/WSOrchestration.pdf

[14] Potts, M. et al., 2003. Web Services Manageability. IBM Corporation, Computer Associates Internatioanl, Inc., and Talking Blocks, Inc. [Online]. Available at: ftp://www6.software.ibm.com/software/developer/library/ws-manage.pdf

[15] Sandsoft, 2002. Component-Based Ontology Development. Available: http://www.sandsoft.com/ontologies.html

[16] Semaview Inc., 2002. Concept to Reality: What the Emerging Semantic Web means to your Business. [Online]. Available at: http://www.semaview.com/d/whitepaper.pdf

[17] Systinet, 2003. Web Services: A Practical Introduction to SOAP Web Services. [Online]. Available at: http://www.systinet.com/download/8c81238168f0e71660ccb3fcf118f7c2/wp_Introduction_to_Web_Services.pdf

[18] World Wide Web Consortium (W3C), 2003. Web Services Architecture, W3C Working Draft. [Online]. Available at: http://www.w3.org/TR/2003/WD-ws-arch-20030514/