World Journal on Educational Technology

2 (2009) 98-109

www.world-education-center.org/index.php/wjet

Semantic web application for the taking of decisions for WBE systems

Rubén Peredo Valderrama [a], Alejandro Canales Cruz [a] *

[a] Researchers of the Centro de Investigación en Computation–IPN, Unidad Profesional Adolfo López Mateos,
Col. Nueva Industrial Vallejo, Del. Gustavo A, Madero, D.F. 07738 México City, México

Received June 21, 2009; revised August 12, 2009; accepted September 28 , 2009

Abstract

In this paper a new architecture for development of Semantic Web Applications for decision taking using the paradigm of Web-Based Education (WBE) is presented. This architecture is based on the IEEE 1484 LTSA (Learning Technology System Architecture) specification, Multi-Agent System (MAS) and the software components named Intelligent Reusable Learning Components Object Oriented (IRLCOO). IRLCOO are a special type of Sharable Content Object (SCO) used like composition units under the Sharable Content Object Reusable Model (SCORM). SCORM is used to create reusable and interoperable learning content. The new architecture is oriented to offer interoperability at level application under the philosophy of Service-Oriented Architecture (SOA).

Keywords: Semantic Web Application; WBE; ADL-SCORM; Multi-Agent System; IRLCOO

1. INTRODUCTION

Tim Berners-Lee commented on the vision: "*The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation*" (Michael, 2003). Tim Berners-Lee has a vision for the

* Rubén Peredo Valderrama. Tel.: +52 55 57296000x56611; fax: +52 55 57296000x56614
E-mail address: peredo@cic.ipn.mx

future of the Web. The first is to build the Web as a more collaborative medium. The second is to build the Web understandable, and thus processable, by machines. Tim Berners-Lee's original vision clearly involved more than retrieving Hypertext Markup Language (HTML) pages from Web servers. There are relationships between resources that are not currently captured on the Web. The technology to capture such relationships is called the Resource Description Framework (RDF). The fundamental idea is to understand that the novel vision encompassed ad-ditional meta data above and beyond what is currently in the Web. This additional meta data is needed for machines to be able to process information on the Web (Gruber, 1993).

The Web is principally considered for humans and not for computer. The Semantic Web (SW) comprehends techniques that promise radically to improve the current Web and its use. The SW technologies propose a novel vision to manage information and processes using the computer, the fundamental principles are the creation and use of semantic meta data (Gruber, 1993).

The IEEE 1484 LTSA standard is a high level architecture for Learning Management Systems (LMS) supported education, learning and training that describes the high level system design and the components of these systems (IEEE 1484.1/D9 LTSA, 2001).

In general terms, the IEEE 1484 LTSA standard is acknowledged into the Web learning community, the purpose of this paper is to show a new Agents and Compo-nents Oriented Architecture (ACOA) and Semantic Web technologies for the WBE systems development based on the IEEE 1484 LTSA standard. This proposal is a response to a group of restrictions and requirements that the actual educational systems have not been able to provide in proper form. Especially the concrete cases of authoring of content and evaluation systems that have tried to resolve the professor's technical deficiencies through of the simplified creation and publication inside a WBE system, but in most cases these systems do not fulfill the educational software current demands. In the technological aspect we have: accessibility, adaptability, flexibility, interoperability, quality, reuse, reduced times of delivery and cost.In the pedagogic aspect we have: the lack of the employment of pedagogic models in the courses development that help in the learner's learning inside a WBE environment.

The IRLCOOs are part of a new Agents and Components Oriented Architecture (ACOA) based on IEEE 1484 LTSA specification (IEEE 1484.1/D9 LTSA, 2001), and open standards such as eXtensible Markup Language (XML) (XML, 2003), as a coding system and to ensure that the learning content is 'interoperable' with various learning management technologies, mainly based on: Advanced Distributed Learning (ADL), and SCORM (ADL, 2004).

The WBE as a mode of study is due to the increase in the number of students and limited learning content resources available to meet a wide range of personal needs, backgrounds, expectations, levels, skills, etc. Consequently, the purpose of the deliv-ery process is extremely important, because it means to produce learning content and to present it to the learner in multimedia form. To develop, revise and upgrade the learning content in an efficient way. The work described in this paper is based on a special type of labeled materials called IRLCOO, developed by Peredo et al (Peredo, 2005). The IRLCOO represent a type of learning content characterized by rich multimedia, high interactivity and intense feedback that is supported by means of a standard interface and functionality.

The paper is organized of the following way: in Section 2, the IEEE 1484 LTSA specification layers is introduced; Section 3 describes ACOA and shown the software development pattern based on IRLCOO (layer 3); in Sections 4 and 5, the authoring and evaluation systems based on our architecture are shown respectively; section 6 shows the platform Semantic Web. Finally, conclusions are discussed.

2. IEEE 1484 LTSA

The Learning Technology Standards Committee of the IEEE Computer Society has proposed the IEEE 1484 LTSA specification (IEEE 1484.1/D9 LTSA, 2001). This Standard is composed by five layers.

The technical description of this architecture is of two types: normative and informative. The normative text is the proposal of this standard that cannot be changed. It should be implemented as is proposed, for this case only the layer 3 (System components) is normative in this standard. The others are informative layers and provide texts with examples, explanations and guides of how the proposals can be imple-mented. Each one of these layers is described below:

1. Learner and Environment Interactions: Concerns the learner's acquisition, transfer, exchange, formulation, discovery, etc. of knowledge and/or information through interaction with the environment.
2. Learner-Related Design Features: Concerns the effect learners have on the design of learning technology systems.
3. System Components: Describes the component-based architecture, as identi-fied in human-centered and pervasive features.
4. Implementation Perspectives and Priorities: Describes learning technology systems from a variety of perspectives by reference to subsets of the system compo-nents layer.
5. Operational Components and Interoperability - codings, APIs, protocols: Describes the generic "plug-n-play" (interoperable) components and interfaces of infor-mation technology-based learning technology architecture, as identified in the stake-holder perspectives.

3. AGENTS AND COMPONENTS ORIENTED ARCHITECTURE

The Industrial Software Revolution is based upon component software engineer-ing. Between the reasons that explain the relevance of the Component-Oriented Pro-gramming (COP) are: the high level of abstraction offered by this paradigm and the current trends for authoring reusable component libraries, which support the devel-opment of applications for different domains. In addition, according to Wang (2005) three major goals pursued by COP are considered: conquering complexity, managing change, and reusability.

According to Szyperski (1998) a software component is "*a unit of composition with contractually specified interfaces and explicit context dependencies. A software component can be deployed independently and is object to composition by third parties.*" Although in most cases this definition is acceptable, its meaning is quite generic, so it is not surprising that the term is used to mean rather different concepts.

Our ACOA is based on layer 3 of IEEE 1484 LTSA specification.This architecture is presented in Fig. 1, and consists of four processes: learner entity, evaluation, coach, and delivery process; two stores: learner records and learning resources; and fourteen information workflows.

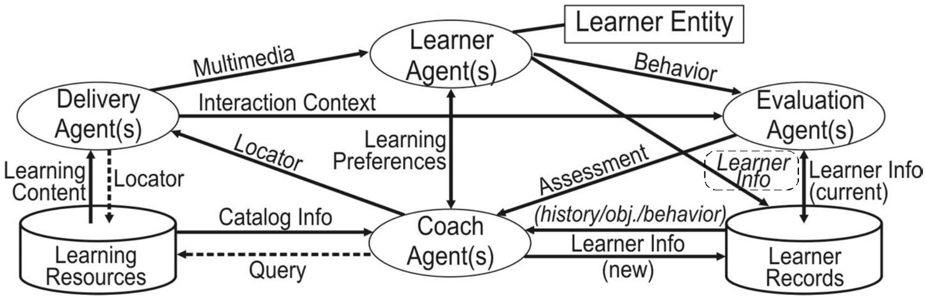

Figure 1. *Components Oriented Architecture*

First, the coach process has been divided in two subprocesses: coach and virtual coach. The reason is because we considered that this process has to adapt to the learner's individual needs in a quick way during the learning process. For this, some decisions over sequence, activities, examples, etc., can be made manually for the coach but in others cases these decisions can be made automatically for the virtual coach.

Briefly, the overall operation has the following form: (1) the learning styles, strate-gies, methods, etc., are negotiated among the learner and other stakeholders and are communicated as learning preferences; (2, new proposal) the learner information (behavior inside the course, e.g., trajectory, times, nomadicity, etc.) is stored in the learner records. The learner information workflow was a new proposal that was not in the architecture original IEEE 1484 LTSA, this new workflow is for the purpose that the components IRLCOO could take metric of the students for a better taking of decisions, registering them in the learner records [10]; (3) the learner is observed and evaluated in the context of multimedia interactions; (4) the evaluation produces assessments and/or learner information; (5) the learner informa-tion (keyboard clicks, mouse clicks, voice response, choices, written responses, etc., all over learner's evaluation) is stored in the learner history database; (6) the coach reviews the learner's assessment and learner information, such as preferences, past performance history, and, possibly, future learning objectives; (7, new proposal) the virtual coach reviews the learner's behavior and learner information, and automatic and swiftly makes dynamic modifications on the course sequence (personalized to learner's needs) based on the learning process design; (8) the coach/virtual coach searches the learning resources, via query and catalog info, for appropriate learning content; (9) the coach/virtual coach extracts the locators (e.g., URLs) from the avail-able catalog info and passes the locators to the delivery process, e.g., a lesson plan or pointers to content; and (10) the delivery process extracts the learning content and the learner information from the learning resources and the learner records respectively, based on locators, and transforms the learning content to an interactive and adaptive multimedia presentation to the learner (Canales, 2007).

3.1. IRLCOO platform

The initial versions of the components IRLCOO were developed with Flash (Adobe, 2005). The client's components allow multimedia content. At Run-Time, the components load media objects and offer a programmable and adaptive environment to the student's necessities. Thus, the components use different levels inside the Flash Player. With this structure, it is possible to generate specialized components which are small, reusable, and suitable to integrate them inside a bigger component at Run-Time. The liberation of ActionScript 3.0 inside Flash CS3 and Flex 3.0 respectively, allows the implementation of the Object Oriented paradigm. With these facilities IRLCOO are tailored to the learner's needs. The last versions of the components IRLCOO were developed with Flex. Flex was released as a J2EE JSP tag library that compiles a tagged-based language called Multimedia eXperience Markup Language (MXML) and an Object Oriented language called Action Script 3.0 straight into Flash applica-tions, which create a binary Small Web File (SWF), applications on the side of the server. The Flex compiler is still a J2EE application but for the client now. A Flex ap-plication uses: prebuilt components, custom components, rich class library, MXML, and ActionScript 3.0. The principal reason for using Flex is that it allows Web applica-tion developers to rapidly and easily build Rich Internet Application (RIA). The IRLCOO is based on the composition pattern, allows building complex systems that are made up of several smaller components.

IRLCOO platform owns certain communication functionalities inside of the Appli-cation Programming Interface developed for the LMS: Multi-Agent System (MAS), and different frameworks, as: AJAX (Grane, 2006), Hibernate (Peak, 2006), Struts (Holmes, 2004), etc., and dy-namic load of Assets in Run-Time.

IRLCOO are meta labeled, which are used to identify the components and to de-termine certain specified characteristics. This contrast is made with the meta labeled Resource Description Framework (RDF) (RDF, 2005), which allows enabling certain grade inferences on the materials by means of the Semantic Web Platform.

3.2. Communication between IRLCOO and Web Services

The WebServiceConnector component and WebServices classes to connect to WS from the IRLCOO were replaced with the mx:WebService component. The mx:WebService component enables the access to remote methods offered by a LMS through SOAP protocol. This gives a WS the ability to accept parameters and re-turn a result to the script; it is possible to access and join information of public and proper WS from IRLCOO components. It is possible to reduce the programming time, since a simple instance of the mx:WebService component is used to make multiple calls to the same functionality within the LMS. The components discover and invoke WS using SOAP and UDDI, via middleware and a JUDDI server (Canales, 2009). Placing a Run-Time layer between a WS client and server dramatically increases the options for writing smarter, more dynamic clients. Reducing the necessity of dependences inside the clients. It is only necessary to use different instances for each one of the different functionalities. WS can be unloaded using the component and deployed within an IRLCOO. The next code shows a request from the learning content to the middle-ware, requesting the WS notes. The answer (URL) is used to call the WS from IRLCOO:

```
// Calling WS from a client IRLCOO - RIA with Flex
<mx:WebService id="WebService"
wsdl=" http://148.204.45.65:8080/juddi/inquiry?wsdl">
<mx:operation name="Notes"
resultFormat="object"
result="Notes_result(event);"
fault="Notes_fault(event);" />
</mx:WebService>
```

3.3. Service-Oriented Architecture

As a complement of the specification in layer 5, we propose the use of SOA. SOA provides services and contents tanks for using the second generation of WS, SOAP, XML, Web Services Definition Language (WSDL) and UDDI. The SOA advantage resides in the services development; they provide interoperability to appli-cation level. This is achieved because the services hide the details of the execution environment, through interfaces that have all the applications. These services are called via WS and they have the ability to emit and consume the data represented with XML, without keeping in mind the development platform, middleware, operat-ing system or hardware type. Besides, SOA permit the integration of existent environments (legacy) (Erl, 2004).

3.4. Semantic Markup

Ontologies serve to standardize and provide interpretations for Web materials. To make materials machine-understandable. Web resources must have semantic markup. Such Web resources enable reasoning about their content and for advanced query answering services. They support ontology creation and maintenance, and help map among different ontologies. The semantic markup is created for the subsystem semantic_markup in an automatically, based on XML and RDF. Fig. 2 shows the meta labeled for IRLCOOs - RIA.

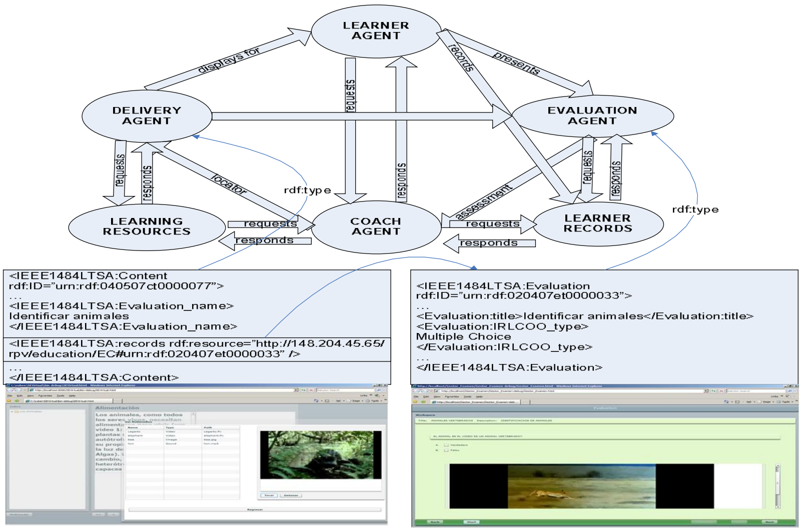

Figure 2. *Meta labeled for IRLCOO – RIA*

3.5. Hibernate

Hibernate is Java middleware designed to achieve the Object Relational Mapping (ORM) model. ORM is the name given to automated solutions to the mismatch prob-lem. The coding SQL is simplified significantly with Hibernate. Hibernate is a non-intrusive solution and is a complete solution to the problem of managing per-sistent data in Java. It mediates the middleware's interaction with our relational data-base, allowing us to concentrate on the real problems. Our middleware require persis-tent data, being one of the fundamental concepts in applications development. The system need preserve data entered by learners via the IRLCOOs when the client is powered off. Object persistence means that individual objects in the client can outlive the application process; they are saved to the Learner Records and recreated at a later point in time. It frees us from tedious tasks. The middleware built with ORM is cheaper,with better performance, less vendor-specific, and more able to cope with changes to the internal object or underlying SQL schema (Peak, 2006). We use Hibernate in most of the subsystems.

4. SiDeC - RIA

In order to facilitate the development of learning content, an authoring system called SiDeC-RIA (Sistema de Desarrollo de eCursos - eCourses Develop-ment System – Rich Internet Application)was built. SiDeC-RIA is a system based on ACOA to facilitate the authoring content to the tutors with multimedia handling. In addition, the Structure and Package of content multimedia is achieved by the use of IRLCOO, as the lowest level of content granularity.

SiDeC - RIA is used to construct Web-based courseware from the stored IRLCOO (Learning Resources), besides enhancing the courseware with various authoring tools. Developers choose one of the SiDeC lesson templates and specify the desired components to be used in each item. The previous version SiDeC supported lesson templates based on the cognitive theory of Conceptual Maps (CM) (Novak, 1984) and Based-Problems Learning (BPL) (Canales, 2007). The SiDec - RIA added new lesson templates additionally based on the: Case Method (CM) (Ellet, 2007), Problem-Based Learning (PBL) (Barell, 2006), Uskov (Vladimir, 2005), and a Structured Open (SO).

The tool supports the generation of IRLCOO - RIA to provide on-line courses. This courseware takes learner's metrics via IRLCOO - RIA with the purpose of tailoring their learning experiences. The IRLCOOs -RIA offer a friendly interface and flexible functionality. These deliverables are compliance with the specifications of the IRLCOO and with SCORM 1.2 Models (Content Aggregation, Sequencing and Navi-gation, and Run Time Environment) (ADL, 2004). Meta data represent the specific description of the component and its contents. The meta data tool provides templates for entering meta data and storing each component in the SiDeC - RIA.

SiDeC – RIA has a course structure based on the idea of a compound learning item as a collection of Reusable Learning Atoms (RLA) and Reusable Information Atoms (RIA) (Vladimir, 2005). These atoms are grouped together to teach a common task based on a single learning objective. A RLA is an elementary atomic piece of learning that is built upon a single learning objective. Each RLA can be classified as: concept, fact, process or procedure. The RLAs provide the information of learner's behavior within the course. This information is stored in the learner history database (Learner Records).

5. Evaluation System - RIA

The Evaluation System – RIA for WBE is designed under the same idea used for the SiDeC - RIA. The functionality of the Evaluation System - RIA lays on the analysis of the learner's profile, which is built during the Teaching/Learning experi-ences. The profile is based on metrics that are extracted from the learner's behavior at Run-Time via the IRLCOOs - RIA. These metrics are stored into the Learner Records that compose the profile. The new sequences of courses is functioned of the results obtained during the course.

The Evaluation System - RIA combines IRLCOOs - RIA, additional meta-labels based on RDF, and the Java Agent Platform. Moreover, technologies of the Artificial Intelligence (AI) field are considered in order to create a Semantic Web environment. The objectives of the Semantic Web are to assist human users to achieve their online activities. Semantic Web offers abundance advantages, such as: reduction of the complexity for potential developers, standardization of functionalities and attributes, definition of a set of specialized APIs, and deployment of a Semantic Web Platform.

The resources have a Universal Resource Identifier (URI). The XML layer is used to define the SCORM meta data of IRLCOO - RIA that are used to interchange data over the Web. The XML Schema layer corresponds to the language used to define the meta data structure (XML, 2003). The RDF layer is represented by the language used for de-scribing all information, meta data, and relationships. The RDF Schema layer is car-ried out by the Framework that provides meaning to the vocabulary implemented (RDF, 2005). The Ontology layer is dedicated to define the semantics for establishing the usage of words and terms in the context of the vocabulary (Passin, 2004 & Antoniou, 2004).

Essentially the Evaluation System - RIA is fulfilled through two phases. The first phase is supported by the LMS, and is dedicated to present the course and its struc-ture. The actions are registered and the presentation of the contents is realized with IRLCOO - RIA content. The evaluations are done by evaluating based on IRLCOO - RIA and in some cases by simulators based on IRLCOO - RIA. These processes are deployed by the Framework of Servlets/Java Server Pages/JavaBeans.

The second phase analyzes the learner's records carried out by the Server based on JADE - JADEX MAS (JADE, 2009 & JADEX, 2009). This agent platform owns seven agents: Snooper, Buffer, Learner, Evaluation, Delivering, Coach, and Info. The primary idea is to automate the learner's analysis through the coach/virtual coach, and to give partial results that can be useful for the learner's final instruction. These agents are imple-mented as Java Beans programs, which are embedded in the applications running both at the client and server sides. The Snooper Agent works as a trigger by means of the INFORM performative, which activates the MAS server's part. This agent is deployed into a Java Server Page that uses a Java Bean. During the lesson or once evaluation is finished, the graphical user interface activates the Snooper Agent and sends the behavior/evaluation metrics (using Agents Communications Language (FIPA XC00061, 2001)) to be analyzed at the server side by the MAS. The Snooper Agent activates the system, whereas the Buffer Agent manages the connection and all the messages from the client. Both tasks are buffered and sent to the Coach Agent. Then the Coach Agent requests to the Learner Records the learner preferences, trajectory, previous learner monitoring information, etc. The Coach Agents analyzes this information to determine if the learner needs help. If the answer is yes, the Coach Agent requests from the Learning Resources the needful learning content (URLs) and it sends the learning contents (URLs) to the Delivery Agent. The Delivery Agent sends the learning content to the Learner and Evaluation Agents for its presentation. These agents employ the dynamic sequencing to change the course or assessment sequence. The sequencing is defined for the instructional strategy based on pedagogical model and employs the SCORM Sequencing/Navigation. Once the necessary information is received (sequence, IRLCOO - RIA type and localization, etc.), this is represented as a string, which is constructed dynamically by the rule-based inference engine known as JENA (JENA, 2009), to generate dynamic feedback.

6. SEMANTIC WEB PLATFORM

The overall architecture of Semantic Web Platform for WBE, which includes three basic engine representing different aspects, is depicted in Fig. 3.

1. The query engine receives queries and answers them by checking the content of the databases that were filled by Coach, Evaluation and Learner agents and the inference engine.

2. The database manager is the backbone of the entire systems. It receives facts from the Coach, Evaluation and Learner agents, exchanges facts as input and output with the inference engine, and provide facts to the query engine.

3. The inference engine use facts and our ontologies to provide additional factual knowledge. It frees knowledge providers from the burden of specifying each fact explicitly.

The ontologies are the overall structuring principle. The Coach and Evaluation agent use them to extracts facts, the inference engine to infer facts, the database manager to structure the database, and query engine to provide help in formulating queries.

JENA was selected as the inference engine. It is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine (JENA, 2009).

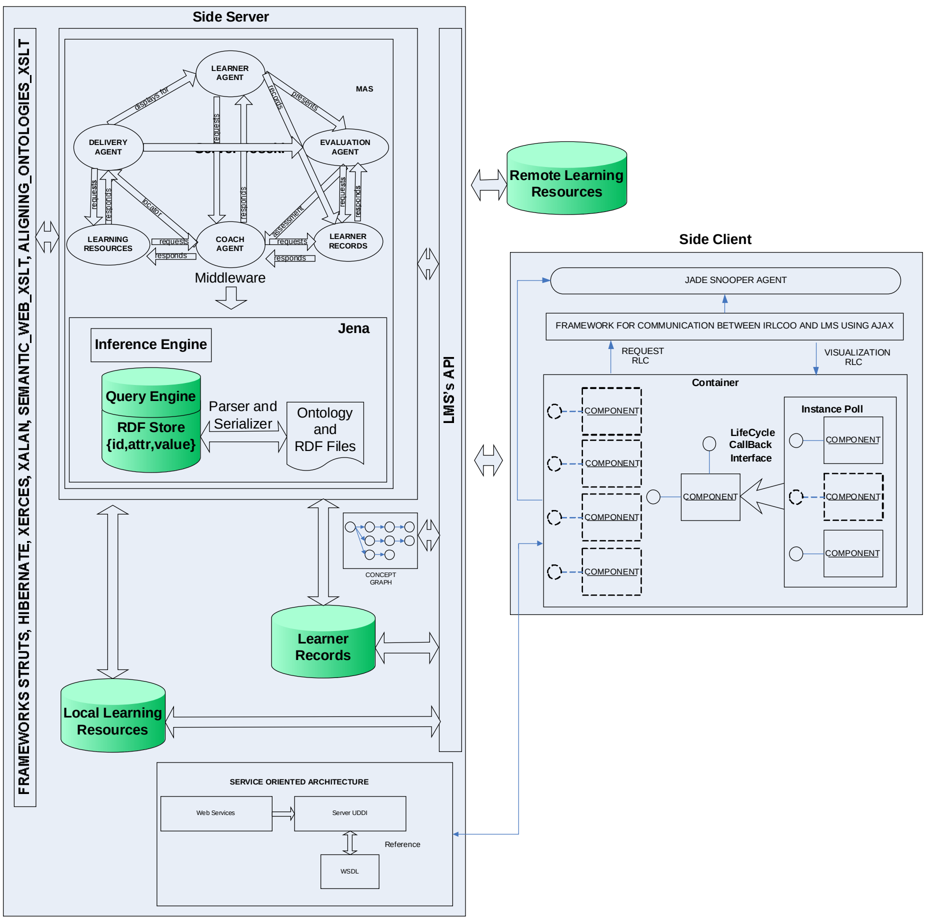

Figure 3. *Semantic Web Platform for WBE*

7. CONCLUSIONS

The SiDec - RIA and Evaluation System - RIA were created under the architecture for adaptive and intelligent materials for WBE. This architecture is integrated for ACOA, JADE-JADEX MAS, IRLCOO - RIA and Semantic Web Platform. Our approach focus on: reusability, accessibility, durability, and, interoperability of the learning contents, which are built as IRLCOO - RIA, as the main component for delivering content and assessment.

Our communication model is composed of the LMS communication API, AJAX, Struts Framework, Hibernate, IRLCOO -RIA, WS, Semantic Web, and a server JUDDI. It provides new development capabilities for WBE systems, because their integrant technologies are complementary. SiDeC - RIA and the Evaluation System - RIA were developed under this model to assist in the automation and reduce of the complexity of the process of developing educational materials.

The incorporation of Web Semantic Platforms assists in creating intelligent and adap-tive systems, according to the learner's necessities.

The ADL-Schema manages dynamic sequencing, composition, content/navigation separation in RTE to develop content and assessments on the Web. ACOA has the same ADL advantages and adds the capacity of generates desk and Web CASE tools using the same content and assessment IRLCOO - RIA. Finally the architecture IEEE 1484 LTSA is modified, for collecting certain student's metric automatically.

Acknowledgments

The authors thank CIC-IPN, to the project SIP-IPN: 20091085 for economic support on the development of this work.

REFERENCES

Adobe (2005). *Adobe Site*. Retrieved 30 July, 2005 from http://www.adobe.com/

ADL (2004). *Advanced Distributed Learning Initiative*. Retrieved 28 August, 2005 from http://www.adlnet.org/Pages/Default.aspx

Antoniou, G. and Van Harmelen, F. (2004). *A Semantic Web Primer*. Cambridge: The MIT Press.

Barell, J. (2006). *Problem-Based Learning: An Inquiry Approach*. Thousand Oaks: Corwin Press.

Canales, A., Peña, A., Peredo, R., Sossa, H. and Gutiérrez, A. (2007). Adaptive and intelligent Web based education system: Towards an integral architecture and framework. *Expert Systems with Applications*, 33(4), 1076-1089.

Canales, A. and Peredo, R. (2009). Adaptive and Intelligent Agents Applied in the Taking of Decisions Inside of a Web-Based Education System. *Springer - Intelligent Agents in the Evolution of Web and Applications. Studies in Computational Intelligent, 167*, 87-112.

Erl, T. (2004). *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. New Jersey: Prentice Hall Editorial.

Ellet, W. (2007). *The Case Study Handbook: How to Read, Discuss, and Write Persuasively About Cases*. Boston: Harvard Business Press.

FIPA XC00061, T. (2001). *FIPA ACL message structure specification*. Retrieved 10 January, 2005 from http://www.fipa.org/specs/fipa00061/

Grane. D., Pascarello, E., and James, D. (2006). *Ajax in Action*. Greenwich: Manning Publications.

Gruber, T. (1993). *A translation approach to portable ontologies. Knowledge Acquisition*. Retrieved 12 August, 2009 from http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html

Holmes, J. (2004). *Struts: The Complete Reference*. California: Mc Graw Hill – Osborne Publications.

IEEE 1484.1/D9 LTSA (2001). *A Draft Standard for Learning Technology - Learning Technology Systems Architecture (LTSA)*. Retrieved 10 January, 2007 from http://ieee.ltsc.org/wg1

JADE (2009). *Java Agent DEvelopment Framework*. Retrieved 10 January, 2009 from http://jade.tilab.com

JADEX (2009). *Jadex BDI Agent System - Overview (About.Overview)*. Retrieved 20 January, 2009 from http://jadex.informatik.uni-hamburg.de/bin/view/About/Overview

JENA (2009). *Jena Semantic Web Framework*. Retrieved 21 January, 2009 from http://jena.sourceforge.net/

Michael C. Daconta, Leo J. Obrst and Kevin T. Smith. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Indianapolis: John Wiley & Sons Publications.

Novak, J. and Gowin, B. (1984). *Learning How to Learn*. Cambridge: Cambridge University Press.

Passin, T. (2004). *Explorer's Guide to Semantic Web*. Greenwich: Manning Publications Co.

Peak, P. & Heudecker, N. (2006). *Hibernate Quickly*. Greenwich: Manning Publications.

Peredo, R., Balladares, L., & Sheremetov, L. (2005). Development of intelligent reusable learning objects for web-based education systems. *Expert Systems with Applications*, 28(2), 273-283.

RDF (2005). *RDF specification*. Retrieved 25 February, 2009 from http://www.w3.org/RDF/

Szyperski, C. (1998). Component Software. Beyond Object-Oriented Programming. New York: Addison-Wesley Editorial.

Vladimir, U., and Maria, U. (2005). Reusable learning objects approach to Web-Based Education. *International Journal of Computer and Applications*, 25(3).

Wang, A. and Qian, K. (2005). *Component-Oriented Programming* . Georgia: Addison-Wesley Editorial.

XML (2003). *Extensible Markup Language, W3 Consortium*. Retrieved 17 March, 2005 from http:/www.w3.org/XML