

Towards an Integral Architecture for Semantic Web, Adaptive and Intelligent WBE Systems

Alejandro Canales Cruz¹, Rubén Peredo Valderrama¹, J. Humberto Sossa Azuela¹
and Agustín Fco. Gutiérrez Tornés²

¹ Computer Science Research Center of National Polytechnic Institute, Mexico City, Mexico

² National Bank of Mexico - BANAMEX, Mexico City, Mexico

+ (52) 55 57296000 Ext. 56611

{acc, peredo, hsossa}@cic.ipn.mx; agutornes@banamex.com.mx

Abstract. A new Agents and Components Oriented Architecture for the development of Web-Based Education systems is presented. This architecture is based on the IEEE 1484 LTSA (Learning Technology System Architecture) specification and the software development pattern of the Intelligent Reusable Learning Components Object Oriented (IRLCOO). IRLCOO are a special type of Sharable Content Object (SCO) according to the Sharable Content Object Reference Model (SCORM). SCORM is used to create reusable and interoperable learning content. The architecture, communication model (LMS API, Web Services, AJAX, and Struts Framework) and Semantic Web Platform that is described in this paper is used to develop authoring and evaluation tools oriented to offer application level interoperability under the philosophy of Web Services.

1 Introduction

The use of 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, skills, levels, etc. Therefore, the purpose of the delivery process is very important, because it means to produce learning content and to present it to the learner in multimedia form. Nowadays, there are approaches over this process that focus on new paradigms to produce and deliver quality content for online learning experiences. These approaches try 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 [1]. The IRLCOO represent a kind of learning content characterized by rich multimedia, high interactivity and intense feedback that is supported by means of a standard interface and functionality.

The IRLCOO are part of a new Agents and Components Oriented Architecture (ACOA) based on IEEE 1484 LTSA specification [2] and open standards such as XML [3] as a bar coding system and to make sure that the learning content is interoperable, the Global IMS Learning Consortium [4], Advanced Distributed Learning (ADL), and SCORM [5]. This paper is organized as follows: in Section 2, ACOA and

IRLCOO are described; in Section 3, the authoring system based on ACOA and Conceptual Maps is presented; furthermore in Section 4, the evaluation system based on ACOA and Semantic Web Platform is showed; finally, the conclusions are discussed.

2 Agents and Components Oriented Architecture

Between the key issues of software engineering is the aim for developing quality software. Thus, components are widely seen by software engineers as a main technology to address the “software crisis.” The Industrial Software Revolution is based upon component software engineering. Between the reasons that explain the relevance of the Component-Oriented Programming (COP) are: the high level of abstraction offered by this paradigm and the current trends for authoring reusable component libraries, which support the development of applications for different domains. In addition, according to Wang [6] three major goals pursued by COP are considered: conquering complexity, managing change, and reusability.

According to Szyperski [7] 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 in four processes: learner entity, evaluation, coach, and delivery process; two stores: learner records and learning resources; and fourteen information workflows.

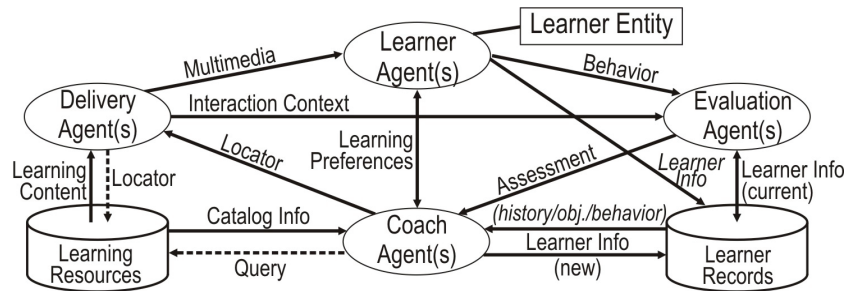


Fig. 1. Components Oriented Architecture.

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 learners’ 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, strategies, 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; (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 information (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 automatically and smartly he 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 available 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. In the section 4 describes ACOA from agents viewpoint.

2.1 IRLCOO platform

IRLCOO were developed with Flash. Flash is an integrator of media and have a powerful programming language denominated ActionScript 2.0 [8]. This language is completely Object Oriented and enables the design of client components that allows multimedia content. At Run-Time, the components load media objects and offer a programmable and adaptive environment to the student's necessities. Flash already has Smart Clips for the learning elements denominated Learning Interactions. The aim is to generate a multimedia library of IRLCOO for WBE systems with the purpose to separate the content from the control. Thus, the components use different levels of code 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 version 2.0 inside Flash MX 2004 allows the implementation of the Object Oriented paradigm. With these facilities IRLCOO are tailored to the learners' needs. In addition, this IRLCOO development platform owns certain communication functionalities inside the Application Programming Interface with LMS, Multi-Agent System (MAS), and different frameworks, as AJAX [9], Hibernate [10], Struts [11], etc.), and dynamic load of Assets in Run-Time.

IRLCOO are meta-labeled with the purpose of complete a similar function as the product bar codes, which are used to identify the products and to determine certain characteristics specify of themselves. This contrast is made with the meta-labeled Resource Description Framework (RDF/XML) [12], which allows enabling certain grade inferences on the materials by means of the Semantic Web Platform.

2.2 Communication between IRLCOO and Web Services

ActionScript 2.0 adds the component `WebServiceConnector` to connect to Web Services (WS) from the IRLCOO. The `WebServiceConnector` component enables the access to remote methods offered by a LMS through SOAP protocol. This gives to a WS the ability to accept parameters and return a result to the script, in other words, it is possible to access and join data between public or own WS and the IRLCOO. It is possible to reduce the programming time, since a simple instance of the `WebServiceConnector` 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. Placing a Run-Time layer between a WS client and server dramatically increases the options for writing smarter, more dynamic clients. Reducing the needs for hard-coded dependencies within WS 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.

3 SiDeC

In order to facilitate the development of learning content, it was built an authoring system called SiDeC (*Sistema de Desarrollo de eCursos - eCourses Development System*). SiDeC is a system based on ACOA to facilitate the authoring content to the tutors who are not willing for handling multimedia applications. In addition, the Structure and Package of content multimedia is achieved by the use of IRLCOO, as the lowest level of content granularity.

SiDeC 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. At this moment, the SiDeC lesson templates are based on the cognitive theory of Conceptual Maps (CM) [13], but in the future we will consider others theories such as: Based-Problems Learning (BPL), the cases method, etc.

The inclusion of cognitive elements, as CM, obeys to the instructional design pattern for the development of the courses. Thus, the courses do not only have theoretical or practical questions, but rather they include a mental model about individual thought process. CM is a schema to structure concepts with the purpose of helping the learners to maximize the knowledge acquisition. A CM is a graphical technique used during the teaching-learning process, among other forms as instructional and learning strategy, and as schematic resource or navigation map.

A metadata tool supports the generation of IRLCOO to provide on-line courses. This courseware estimates learners' metrics with the purpose to tailor their learning experiences. Furthermore, the IRLCOO offer a friendly interface and flexible functionality. These deliverables are compliance with the specifications of the IRLCOO and with learning items of SCORM 1.2 Models (Content Aggregation, Sequencing and Navigation, and Run Time Environment) [5]. Metadata represent the specific description of the component and its contents, such as: title, description, keywords,

learning objectives, item type, and rights of use. The metadata tool provides templates for entering metadata and storing each component in the SiDeC or another IMS/IEEE standard repository.

SiDeC proposes 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) [14]. These atoms are grouped together to teach a common task based on a single learning objective, as is depicted in Fig. 2. 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, e.g., trajectory, times, and assessments. This information is stored in the learner history database (learner records).

On the other hand, a RIA is an atomic piece of information that is built upon single information object. It may contain up to seven different content items, such as: overview, introduction, importance, objectives, prerequisites, scenario, and outline.

In Fig. 2, the SiDeC implements the CM as a navigation map or instructional and learning strategy allowing to the learner to interact with content objects along the learning experiences. These experiences follow an instructional-teaching strategy. These kinds of strategies carry out modifications of the learning content structure. Such modifications are done by the designer of the learning experience with the objective of provide significant learning and to teach the learners how to think [15]. The learning content can be interpreted in a Learning Content Tree.

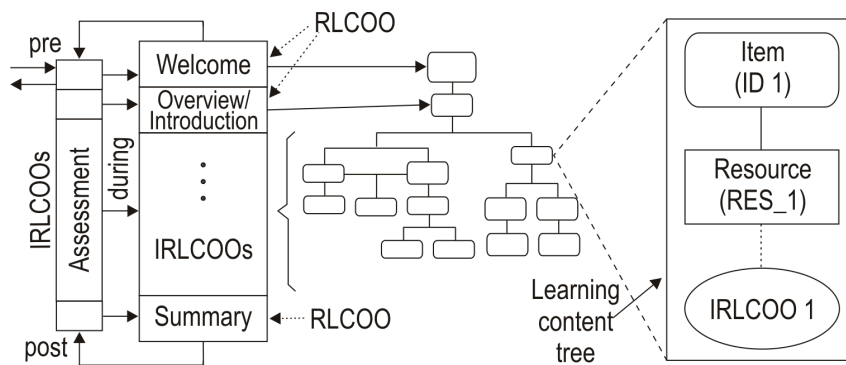


Fig. 2. Learning content generated for the SiDeC.

3.1 Communication between IRLCOO and LMS

Our communication model uses an asynchronous mode in Run-Time Environment (RTE) and joins to LMS communication API of ADL [5], and AJAX (Asynchronous JavaScript And XML) [9] and Struts Framework [11] for its implementation. The LMS communication API of ADL consists of a collection of standard methods to let the Client to communicate with the LMS.

AJAX is a Web development technique to create interactive applications that it is executed in client side, in other words, the Web browser maintains the asynchronous

communication with the server in backstage. This way it is possible to carry out changes in the same page without necessity of reload it. This increases the interaction speed.

On the other hand, the Struts Framework is a tool for Web application development under the Java MVC (Model-View-Controller) architecture; with this Framework is defined the independent implementation of the Model (business object), the View (in-terface with the user or another system) and the Controller (controller of the application workflow). This Framework provides the advantage of maintainability, performance (tags pooling, caching, etc.), and reusability (contains tools for the field validation that it is executed in client or server sides). The browser-based communication model is depicted in Fig. 3.

According to Fig. 3, the communication model starts: (1) when an IRLCOO generates an event. (2) Form the browser interface is made a JavaScript call to the function `FileName_DoFSCommand(command,args)`, which handles all the FSCommand messages from IRLCOO, LMS communication API, and AJAX and Struts methods. Next, a fragment of this code is showed:

```
function FileName_DoFSCommand(command,args) {
    doInitialize();

    doSetValue(name,value); // i.e. (StudentName,name)

    doTerminate();

    useAjaxStruts(); }
```

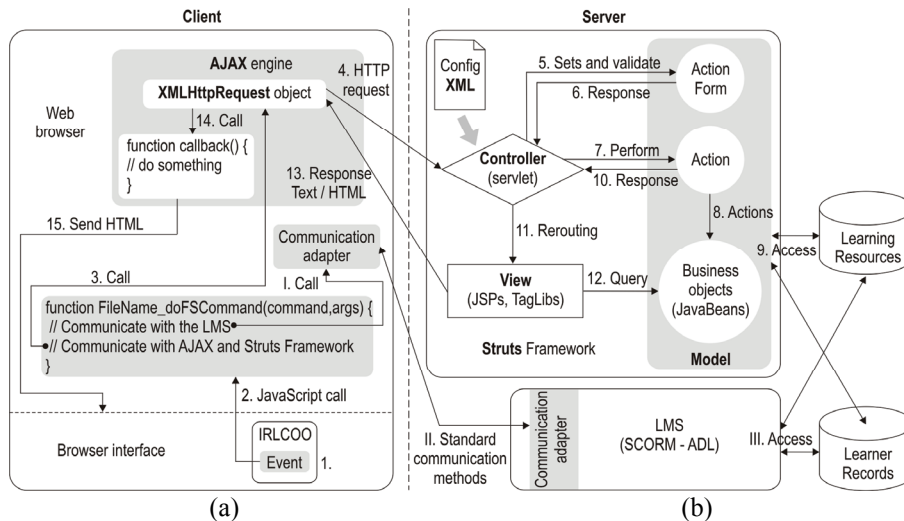


Fig. 3. Communication model between IRLCOO, LMS and AJAX and Struts Framework.

The communication with the LMS starts when: (I) the standard methods call to the Communication Adapter (written in JavaScript). (II) The communication adapter implements the bidirectional communication ADL's API between the Client and the LMS. (III) The LMS realizes the query-response handling and the business logic, i.e., the access to the database.

The communication with AJAX and Struts Framework begins when AJAX-Struts method is called. (3) An instance of the XMLHttpRequest object is created. Using the open() method, the call is set up, the URL is set along with the desired HTTP method, typically GET or POST. The request is actually triggered via a call to the send() method. This code might look something like this:

```
function useAjaxStruts () {
    createXMLHttpRequest();
    var url = "register.do?";
    var urlString = createUrlString();
    xmlhttp.open("POST",url,true);
    xmlhttp.onreadystatechange = processStateChange;
    xmlhttp.setRequestHeader("Content-Type",
        "application/x-www-form-urlencoded;");
    xmlhttp.send(urlString); }
```

(4) A request is made to the server, this might be a call to a servlet or any server-side technique. (5) The Controller is a servlet which coordinates all applications activities, such as: reception of user data, (6) data validations, and control flow. The Controller is configured for a XML file. (7) The Controller calls to perform method of Action, it passes to this method the data values and the Action reviews the characteristic data that correspond to the Model. (8) The business objects (JavaBeans) realize the business logic, (9) usually a database access. (10) The Action sends the response to the Controller. (11) The Controller reroutes and generates the interface for the results to the View (JSPs). (12) The View makes the query to the Business objects based on the correspondent interface. (13) The request is returned to the browser. The Content-Type is set to text/xml, the XMLHttpRequest object can process results only of the text/html type. In more complex instances, the response might be quite involved and include JavaScript, DOM manipulation, or other technologies. (14) The XMLHttpRequest object calls the function callback() when the processing returns. This function checks the readyState property on the XMLHttpRequest object and then looks at the status code returned from the server. (15) Provided everything is as expected, the callback() function sends HTML code and it does something interesting on the client, i.e. advanced dynamic sequence.

This communication model provides new wide perspectives for the WBE systems development, because it provides the capabilities of communication, interaction, interoperability, security, and reusability, between different technologies. For example, the LMS communication API allows us to make standard database queries of learners' information such as personal information, scores, assigned courses, trajectory, etc. While the communication with AJAX and Struts Framework provides the capability of modify the learner's trajectory according to variables from the learner records in RTE (advanced dynamic sequence), components management (IRLCOO) – remember that these components are built and programming with XML – then, this model provides the way to write, load, change and erase XML files in the Server side.

4 Evaluation System

The Evaluation System for WBE is designed under the same philosophy used for the SiDeC. The functionality of the Evaluation System lays on the analysis of the learner's profile, which is built during the teaching-learning experiences. The profile is based on metrics that elicited from the learner's behavior at Run-Time. These measures are stored into the learner records that compose the profile. The generation of new sequences of courses is in function of the results obtained, besides the account of the adaptation level.

The Evaluation System combines IRLCOOs, additional meta-labels, and a Java Agent platform. Also, some technologies of the Artificial Intelligence field are considered in order to recreate a Semantic Web environment. Semantic Web aims for assisting human users to achieve their online activities. Semantic Web offers plenty of 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.

All resources have a Universal Resource Identifier (URI). An URI can be a Unified Resource Locator (URL) or some other type of unique identifier. An identifier does not necessarily enable access to a resource. The XML layer is used to define the SCORM metadata of IRLCOO that are used to interchange data over the Web. XML Schema tier corresponds to the language used to define the structure of metadata [3]. RDF level is represented by the language used for describing all information and metadata sorts. RDF Schema layer is carried out by the Framework that provides meaning to the vocabulary implemented [12]. The Ontology tier is devoted to define the semantics for establishing the usage of words and terms in the context of the vocabulary [16]. Logical level corresponds to the reasoning used to establish consistency and correctness of data sets and to infer conclusions that are not explicitly stated [17]. The Proofs layer explains the steps of logical reasoning. The Trust tier provides authentication of identity and evidence of the trustworthiness of data, services and agents.

In resume, the components and operation of the SiDeC and Evaluation System are outlined in Fig. 4. Basically the Evaluation System is fulfilled through two phases. The first phase is supported by the LMS, and is devoted to present the course and its structure. All the actions are registered and the presentation of the contents is realized

with IRLCOO content. The evaluations are done by evaluating IRLCOO and in some cases by simulators based on IRLCOO. These processes are deployed by the Framework of Servlets, Java Server Pages and JavaBeans.

The second phase analyzes the learner's records carried out by the Server based on JADE MAS. This agent platform owns seven agents: Snooper, Buffer, Learner, Evaluation, Delivering, Coach, and Info. The fundamental 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 implemented as JavaBeans 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 JavaBean. During the lesson or once evaluation is finished, the graphical user interface activates the Snooper Agent and sends it the behavior or evaluation metrics (using Agents Communications Language [18]) to be analyzed at the server-side of 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 send them to the Coach Agent. Then the Coach Agent requests to the learner records for the preferences learner, trajectory, previous learner monitoring information, etc. The Coach Agents analyzes this information to determine if the learner needs help. If this situation is true, the Coach Agent requests to 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 CM and it employs the SCORM Sequencing/Navigation. Once the necessary information is received (sequence, kind of IRLCOO and localization, etc.), this is represented as a string, which is constructed dynamically by the rule-based inference engine known as JENA [19] and JOSEKI server [20], to generate dynamic feedback.

Fig. 5 illustrates an overview of how our systems are integrated with ACOA. In general, the software architecture is divided into four layers: application, agents & components, database, and server layers. The application layer includes an administration system, which is the ADL platform, to allow system administrators, instructors, and learners to manage learner records and curriculum. On the left side below the administration system, asynchronized ours systems are incorporated. Thus, structure authoring systems are separated from learning content.

4.2 Semantic Web Platform

The overall architecture of Semantic Web Platform, which includes three basic engine representing different aspects, is provided in Fig. 4.

1. The query engine receives queries and answers them by checking the content of the databases that were filled by info agent and inference engine.

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

3. The inference engine use facts and ontologies to derive additional factual knowledge that is only provided implicated. It frees knowledge providers from the bur-den of specifying each fact explicitly.

Again, ontologies are the overall structuring principle. The info agent uses 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 [19].

While JOSEKI was selected as Web API and server. It is an HTTP and SOAP engine supports the SPARQL Protocol and the SPARQL RDF Query language. SPARQL is developed by the W3C RDF Data Access Working Group [20].

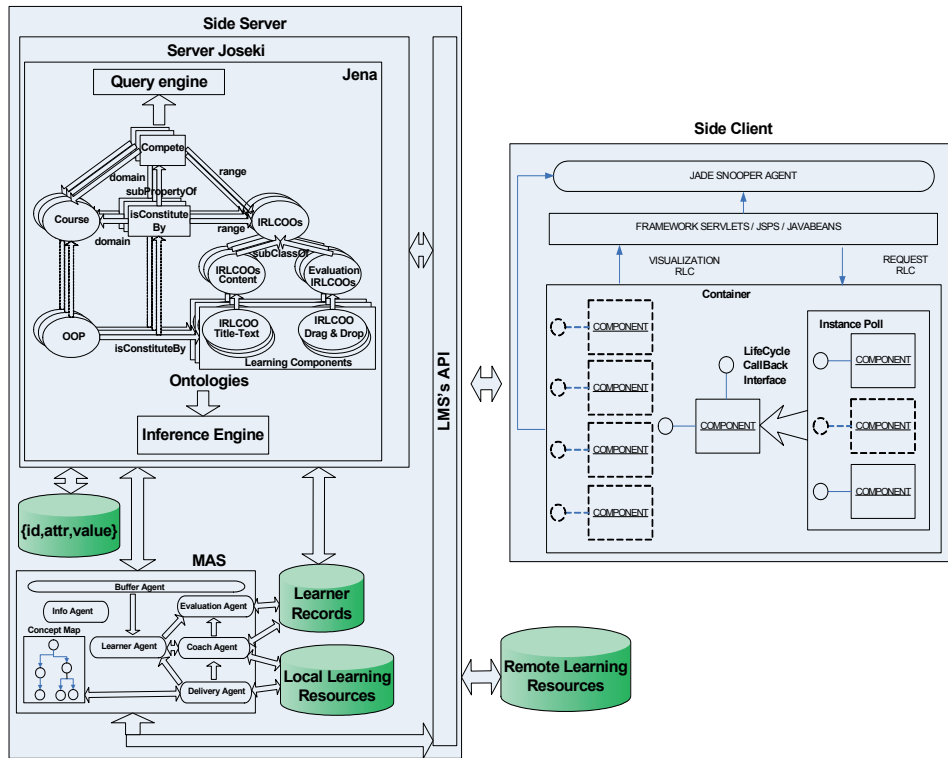


Fig. 4. Semantic Web Platform for WBE.

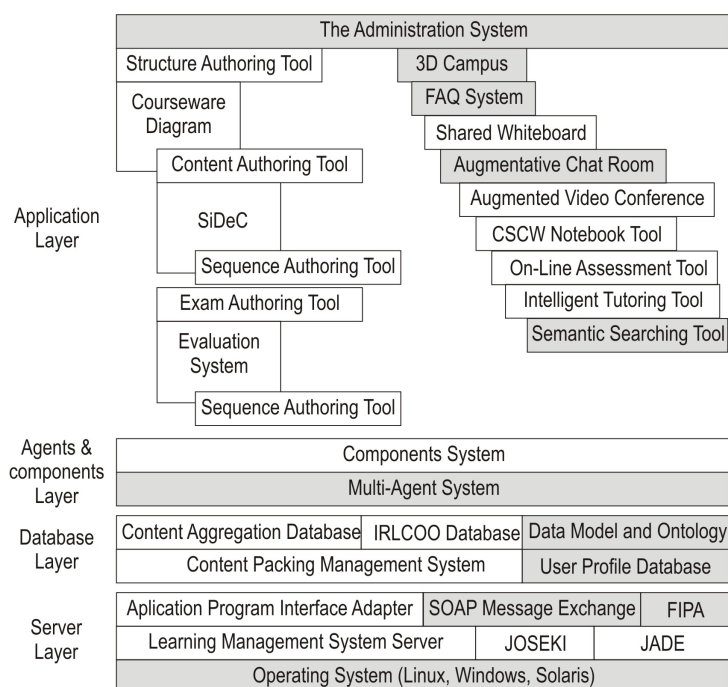


Fig. 5. An integral architecture for WBE systems.

5 Conclusions

ACOA, IRLCOO and Semantic Web Platform allow developing authoring and evaluation systems to create adaptive and intelligent WBE. Our approach focus on: reusability, accessibility, durability, and, interoperability of the learning contents, which are built as IRLCOO, as the main component for delivering learning and evaluation content.

The communication model composes for the LMS communication API, AJAX and Struts Framework, IRLCOO, WS, Semantic Web, and JUDDI. It provides new development capabilities for WBE systems, because their integrant technologies are complementary. SiDeC and the Evaluation System were developed under this model to help in the automation and reduce of the complexity of the learning content process.

The incorporation of Web Semantic Platforms helps us to create intelligent and adaptive systems (bidirectional communication), according to the users needs.

The ADL-Schema manages dynamic sequencing, composition, content separation, and navigation in RTE for development learning and evaluation content in Web. While, our ACOA has the same ADL advantages and adds the capacity of generates desk and Web CASE tools using the same learning and evaluation components generated.

References

1. Peredo, R., Balladares, L., & Sheremetov, L.: Development of intelligent reusable learning objects for web-based education systems. *Expert Systems with Applications*. 28(2). (2005) 273-283
2. IEEE 1484.1/D9 LTSA: Draft Standard for Learning Technology - Learning Technology Systems Architecture (LTSA). New York, USA (2001). URL: <http://ieee.ltsc.org/wg1>
3. XML specifications (2006). URL: <http://www.w3.org/XML/>
4. Global IMS Consortium (2005). URL: <http://www.imspj.org>
5. Advanced Distributed Learning Consortium (2006). URL: <http://www.adlnet.org>
6. Wang, A., & Qian, K.: *Component-Oriented Programming*. John Wiley & Sons, Inc., Publication. Georgia, USA. (2005) 3-5
7. Szyperski, C.: *Component Software. Beyond Object-Oriented Programming*. Addison-Wesley Editorial. USA (1998)
8. Macromedia, Inc. (2006). URL: <http://www.macromedia.com>
9. Grane. D., Pascarello, E., & James, D.: *Ajax in Action*. Manning Publications. Greenwich, USA (2006)
10. Peak, P. & Heudecker, N.: *Hibernate Quickly*. Manning Publications. Greenwich, USA (2006)
11. Holmes, J.: *Struts: The Complete Reference*. Mc Graw Hill – Osborne Publications. Edited by Herbert Schild. California, USA (2004)
12. RDF specification (2005). URL: <http://www.w3.org/RDF/default.htm>
13. Novak, J. & Gowin, B.: *Learning How to Learn*. Cambridge University Press. Cambridge, USA (1984)
14. Vladimir, U., & Maria, U.: Reusable learning objects approach to Web-Based Education. *International Journal of Computer and Applications*, 25(3). (2003)
15. Díaz-Barriga, F.: *Educational strategies for a significant learning*. Mc Graw Hill Publication. Second edition. D.F., México (2002)
16. Passin, T.: *Explorer's Guide to Semantic Web*. Manning Publications Co. USA (2004)
17. Antoniou, G. & Van Harmelen, F.: *A Semantic Web Primer*. The MIT Press. USA (2004)
18. FIPA XC00061 (2001). FIPA ACL message structure specification
19. JENA (2006). URL: <http://jena.sourceforge.net/>
20. JOSEKI server (2006). URL: <http://www.joseki.org/>