An Approach toward a Software Factory for the Development of Educational Materials under the Paradigm of WBE

Rubén Peredo Valderrama
Superior School of Computer Sciences of National Polytechnic Institute, Mexico City, Mexico

peredo@cic.ipn.mx

Alejandro Canales Cruz
National Autonomous University of Mexico, Mexico City, Mexico

alejandro_canales@cuaed.unam.mx

Iván Peredo Valderrama
Superior School of Computer Sciences of National Polytechnic Institute, Mexico City, Mexico

Abstract

The Software Factory described in this paper is an organizational structure specialized in the production of educational materials based on software components according to educational specifications and externally-defined by the end-users requirements (Learners, Tutors, University, Instructional Designers, etc.). The Software Factory applies manufacturing techniques and principles of Domain Engineering to software development, to mimic the benefits of traditional manufacturing. The software components obtained are called Intelligent Reusable Learning - Components Object Oriented (IRLCOO), which are a special type of Sharable Content Object (SCO) according to the Sharable Content Object Reference Model (SCORM). These software components consume Web Services (WS) to produce reusable and interoperable learning content.

Keywords: Software Factory, WBE, IRLCOO, SCORM.

Introduction

The Industrial Revolution created a high expectation in all aspects of society; many industries experienced a dramatic transformation in their methods and tools. Possibly, the automotive industry is the most notable example; the cars are end products, assembled from components supplied by a large number of suppliers, with highly specialized methods and tools. But curiously, software engineering seems not to have made use of the latest technological advances in software production. The programmers have tools, such as

Editor: Janice Whatley

computers, Integrated Development Environment (IDE), libraries, and frameworks, for doing crafted work. But an industry based on craftsmanship is restricted by its methods and tools, and the means of production are overrun. The software is a generic meta product that can be used to create entire families of instances of similar software (Szyperski, 1998). The software industry confronts a demand on a global scale for meta products of high level, needing an innovation technology platform to face a growing demand. Critical innovation is searching how to resolve two chronic problems: complexity and change. There are four areas to resolve the two previous problematic ones: systematic reuse, development assembly, model-driven development, and process frameworks (Greenfield, Short, Cook, Kent, & Crupi, 2004). The term Software Factory has been used to describe large commercial efforts to automate software development along similar lines. There are different definitions of the term software factory, and the following has been chosen as a working definition: "*A software factory is a software product line that configures extensible tools, processes, and content using a software factory template based on a software factory schema to automate the development and maintenance of variants of an archetypical product by adapting, assembling, and configuring framework-based components*" (Greenfield et al., 2004, p. 163). The Software Factory is a class of software production lines to build production lines of concrete products. The main objective is to produce custom applications from a set of reusable components, starting from scratch.

Background

In traditional education students can interact reciprocally in conjunction with the teacher. Unfortunately, the costs of traditional education have been increasing and management of learning resources is not very flexible; another disadvantage of this type of paradigm is that students cannot advance at their own pace of learning. Virtual Education is seen as a good option to the population growth and information, conducting an intensive search for technological/pedagogical solutions in the teaching/learning process. The urgent global requirement of learning resources, reusable, practical and with high quality, involves learning technologies in the process of delivery and innovation, these topics being of high priority in the development of Web Based Education (WBE) systems. The research in WBE systems is centered on accessibility, durability, interoperability and reusability of didactic materials and environments of virtual education. The main initiatives are: Open Knowledge Initiative (OKI) of the MIT (http://www.okiproject.org/), Advanced Distributed Learning (ADL) initiative (http://www.adlnet.gov/Pages/Default.aspx) and the Global IMS Learning Consortium (http://www.imsproject.org). The main initiative is the ADL by standards and tools.

The Semantic Web (SW) comprises techniques that promise to improve radically the current Web. The Web is primarily designed for human beings and not for computers. The SW technologies propose a new vision for managing information and processing using the computer: the basic principle is the creation and use of semantic meta data. The core of SW is the use of ontologies, defined as, "*Ontology is an explicit and formal specification of a conceptualisation of a domain of interest*" (Gruber, 1993. p. 1).

The IEEE 1484 LTSA specification is a high level architecture for information technology-supported learning, education, and training systems that describes the high-level system design and the components of these systems (Learning Technology Standards Committee, 2001).

In general, this standard is accepted into the WBE community and the purpose of this paper is to show an Agents and Components Oriented Architecture (ACOA) for the WBE systems development based on the IEEE 1484 LTSA specification, using ACOA components in the production line to assemble learning resources. This idea arises as an answer to a group of necessities and limitations that the current educational systems have not been able to satisfy in an appropriate form. Such are the cases of the authoring and evaluation system that have tried to solve the professor’s technical deficiencies during the creation and publication of courses inside a WBE system, but in most cases these systems do not fulfill the educational software current demands. In the technological aspects we have the quality, accessibility, flexibility, reuse, adaptability, interoperability, reduction in delivery times and high cost. On the side of the pedagogical aspects we have a lack of pedagogical models in the courses development (learning content) that help in the learner’s learning inside a WBE environment.

The Intelligent Reusable Learning - Components Object Oriented (IRLCOO) are part of ACOA based on IEEE 1484 LTSA specification (Learning Technology Standards Committee, 2001), using open standards such as XML (World Wide Web Consortium, 2006) and Resource Description Framework (RDF) (World Wide Web Consortium, 2010), ensuring that learning content is ‘interoperable’ with various learning management technologies, mainly based on ADL and SCORM (http://www.adlnet.gov/Pages/Default.aspx).

The WBE paradigm has grown in recent years due to the increase in the number of students and available learning resources suitable for a wide range of personal needs, backgrounds, expectations, skills, levels, etc. Therefore, the delivery process is very important, because it produces learning content and presents it to the learner in multimedia format. Nowadays, there are approaches to 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, Balladares, & Sheremetov (2005). The IRLCOO represents a kind of learning content characterized by rich multimedia, high interactivity, and intense feedback supported by a standard interface and functionality.

ACOA proposes enhancement in layers 1, 3 and 5 of the IEEE 1484 LTSA specification. The paper is organized as follows: in the next section, the IEEE 1484 LTSA specification layers are introduced; the third section describes ACOA and shows the software development pattern based on IRLCOO (layer 3); the fourth section describes SiDeC 2.0, Evaluation System 2.0, and VoiceXML based on our architecture are shown respectively; the fifth section depicts the Semantic Web Platform; the sixth section depicts the Software Factory for the development of educational materials. Finally, conclusions are presented.

IEEE 1484 LTSA

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

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 identified 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 components layer.

5. Operational Components and Interoperability - codings, APIs, protocols: Describes the generic "plug-n-play" (interoperable) components and interfaces of information technol-

ogy-based learning technology architecture, as identified in the stake-holder perspectives.

Agents and Components Oriented Architecture

C. Szyperski (1998) defines a software component as "a unit of composition with contractually specified interfaces and explicit context dependencies. A software component can be deployed independently and is subject to composition by third parties" (p. 41). The components are widely seen by software engineers as a main technology to address the so called "software crisis". The Software Industrial Revolution is based upon Component Based Software Engineering (CBSE).

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. Additionally according to Wang and Qian (2005) three major goals pursued by COP are considered: Conquering complexity, Managing Change, and Reusability. The components are used as composition units, adapting, assembling, and configuring. The tools developed are used to create new components or update them.

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

The coach process has been divided into two subprocesses: coach and virtual coach. This was done because we considered that this process has to adapt to the learner's individual needs in a responsive way during the learning process. A number of 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 (Canales, Peña, Peredo, Sossa, & Gutiérrez, 2007).

IRLCOO platform

Flash is a multimedia platform with a powerful programming language denominated ActionScript (AS) 3.0 (Learn ActionScript, 2009). This language is completely Object Oriented and enables the design of client components that allow multimedia content. At Run-Time it loads multimedia components and offers a programmable and adaptive environment for the student's needs. The components use different levels inside the Virtual Machine based on Flash Player (FP). The IRLCOO are tailored to the learner's needs in a number of ways. The last versions of the components were developed with Flex and AS 3.0. Flex was released as a J2EE JSP tag library that compiles a tagged-based language called MXML and an Object Oriented language called AS 3.0 straight into Flash applications, which create binary SWF applications in the side of the server. The Flex compiler is still a J2EE application but for the client now. A Flex application uses prebuilt components, custom components, rich class library, MXML, and AS 3.0. Flex is a cross-platform for the development and deployment of Rich Internet Application (RIA) based on the Adobe Flash platform.

The IRLCOO are based on the composition pattern, to build complex systems that are constituted of several smaller components. This allows developers to manage the components through a common interface. The IRLCOO platform possesses common prebuilt communication, based on the following Application Programming Interfaces (API) developed for the Learning Management System (LMS): Multi-Agent System (MAS), and different frameworks such as AJAX (Grane, Pascarello, & James, 2005), Hibernate (Peak & Heudecker, 2005), Struts (Holmes, 2006), etc., and dynamic load at Run-Time.

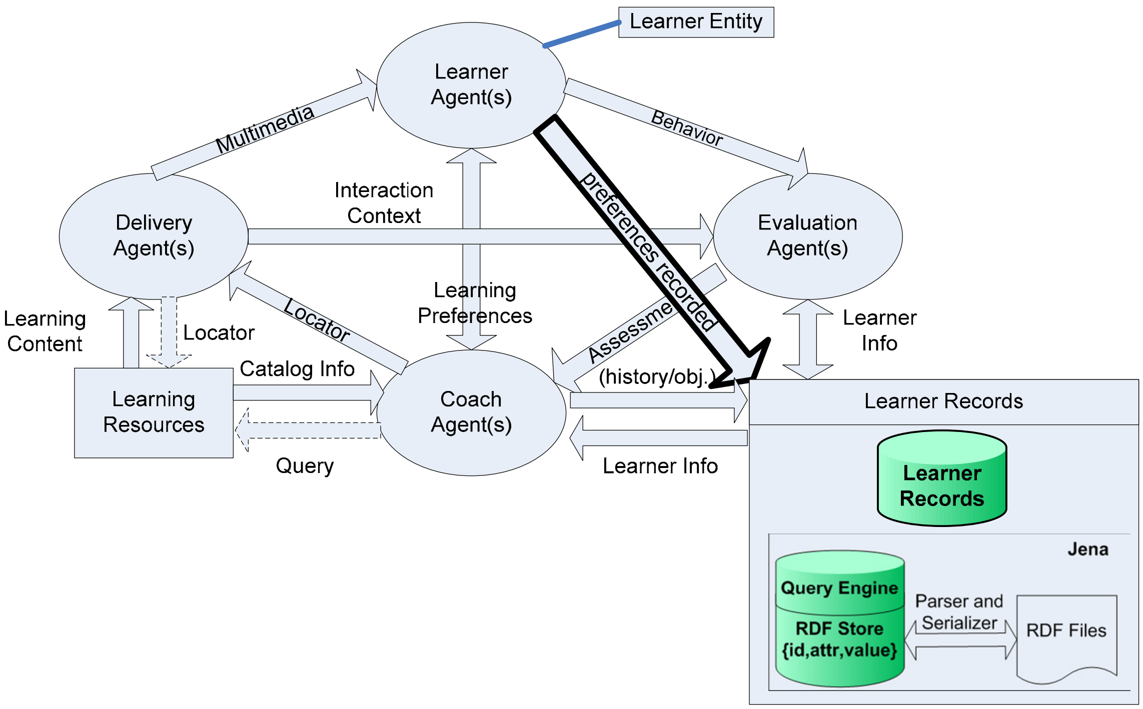


Figure 1. Components Oriented Architecture.

IRLCOO and Web Services

The WebService component gives access to the operations of SOAP-compliant Web Services from the IRLCOO. The component enables access to remote methods offered by a LMS through Simple Object Access Protocol (SOAP) protocol. This gives to a WS the ability to accept parameters and return a result to the script from IRLCOO. The components discover and invoke WS using SOAP and Universal Description, Discovery, and Integration (UDDI) via middleware and a JUDDI server. Placing a middleware layer between a consuming client and a server provider dramatically increases the options to write more dynamic clients, reducing the need for dependences inside the clients. The following code shows a request for the “Notes” from the learning content to the middleware. The WebService prebuilt component is used to invoke the Web Services based on the industry standard SOAP message (World Wide Web Consortium, n.d.). Finally the communication API ADL-SCORM consists of a collection of standard methods to communicate the client with the LMS (http://www.adlnet.gov/Pages/Default.aspx), the standard methods call to the Communication Adapter based on JavaScript, the IRLCOO use the ExternalInterface class to call the Communication Adapter. The answer (URL) is used to call the WS from IRLCOO using the WebService prebuilt component:

```
// Calling WS from a client IRLCOO 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>.
```

Patterns used in the Software Factory

Patterns are forms for encapsulating knowledge with a strategy for solving a frequently occurring problem in a specific context. The pattern defines the relationship between a problem domain and a solution domain.

Composition pattern for IRLCOO

The IRLCOO type components were built using the composition pattern that provides a common interface to manage indivisible and compound components. The IRLCOO use a common interface via a Container_IRLCOO class; the class is defined as an abstract class, but AS 3.0 does not support abstract classes, whereby we define classes knowing that these will not be instantiated, but will be extended by subclasses. Abstract methods will be defined as a function declaration that will throw an error if it is called. The Container_IRLCOO class defines default implementations for both Compound_IRLCOO and indivisible_IRLCOO and overrides necessary methods. The methods will allow the clients to build the compound system. The methods from the base class are not relevant to the indivisible components IRLCOO, nevertheless, are overridden and implemented. The Container_IRLCOO is composed of indivisible and compound IRLCOO. The classes provide a recursive implementation of the iterator() method. Figure 2 shows the class diagram of the composition pattern based on IRLCOO.

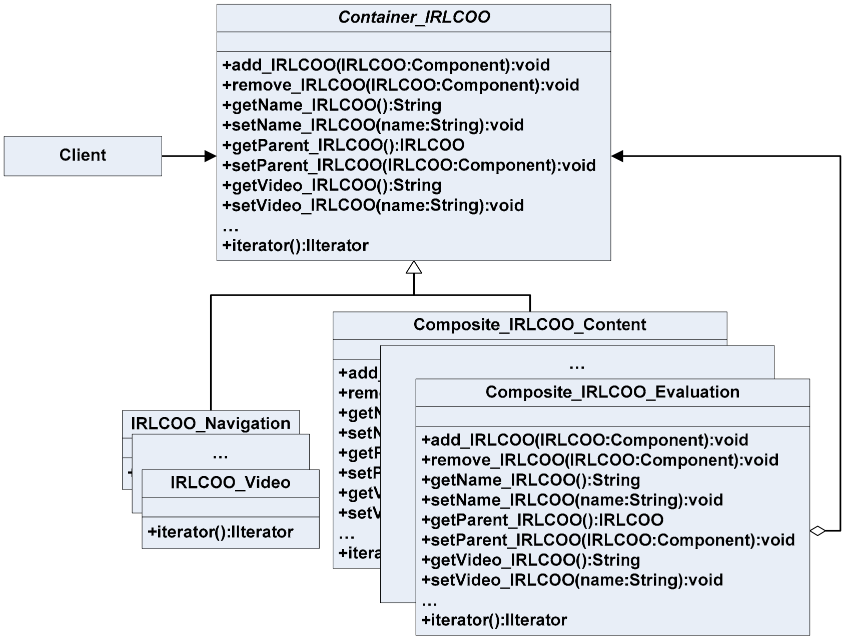

Container_IRLCOO
+add_IRLCOO(IRLCOO:Component):void
+remove_IRLCOO(IRLCOO:Component):void
+getName_IRLCOO():String
+setName_IRLCOO(name:String):void
+getParent_IRLCOO():IRLCOO
+setParent_IRLCOO(IRLCOO:Component):void
+getVideo_IRLCOO():String
+setVideo_IRLCOO(name:String):void
...
+iterator():IIterator
Client
Composite_IRLCOO_Content
...
Composite_IRLCOO_Evaluation
+add_IRLCOO(IRLCOO:Component):void
+remove_IRLCOO(IRLCOO:Component):void
+getName_IRLCOO():String
+setName_IRLCOO(name:String):void
+getParent_IRLCOO():IRLCOO
+setParent_IRLCOO(IRLCOO:Component):void
+getVideo_IRLCOO():String
+setVideo_IRLCOO(name:String):void
...
+iterator():IIterator
IRLCOO_Navigation
...
IRLCOO_Video
+iterator():IIterator

Figure 2. Class diagram of the composite pattern based on IRLCOO.

Model-View-Controller pattern with IRLCOO

The Model-View-Controller (MVC) pattern separates three elements without overlapping every responsibility. The MVC is a compound pattern that consists of several patterns; in our case we use two basic patterns: the observer and composition. The observer pattern is used in order to keep the view updated, as well as the composition pattern, covered in the previous section. Figure 3 shows MVC architecture implemented with Struts 2.0 (http://struts.apache.org/).

Observer pattern with IRLCOO

The Observer pattern has an object called the subject; the subject maintains a list of subscribers, automatically notifying the state change. The IRLCOO components subscribe then begin the service and continue until an IRLCOO component unsubscribes via the WebService component that sends out the state, setting up the IRLCOO component to receive the information from a single source. Figure 3 shows the Observer Pattern implemented.

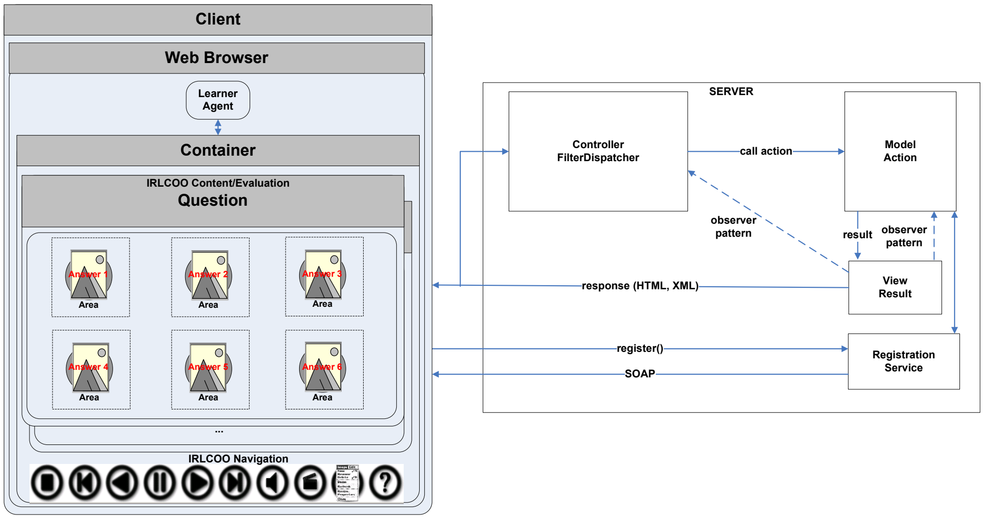


Figure 3. The MVC Architecture implemented with Struts 2 and IRLCOO.

Meta data with IRLCOO

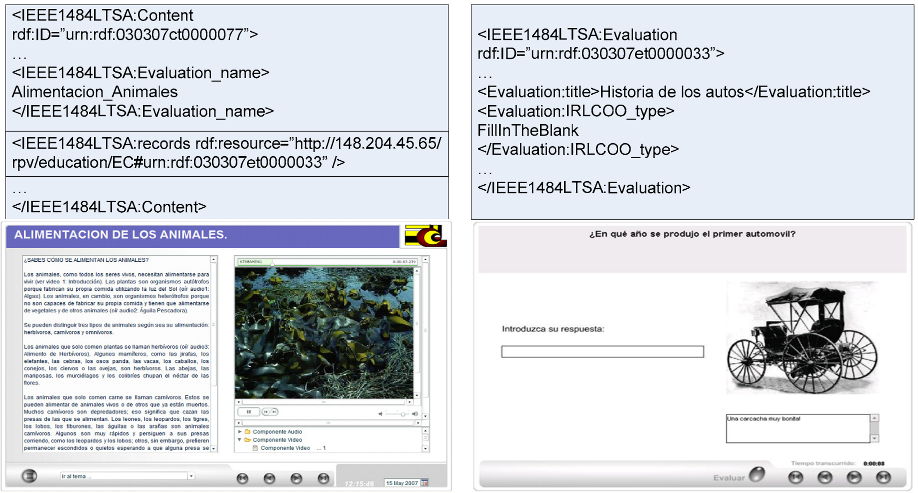
The Ontologies standardize and provide meaning to our materials under the WBE paradigm, enabling understandable educational materials for the machine. The IRLCOO componets have semantic markup and enable reasoning based on the JENA inference engine. The ontology's middleware supports creation and maintenance. The semantic markup is created for the subsystem semantic_markup in an automatic way, based on XML and RDF. Figure 4 depicts the IRLCOO with meta data.

Systems Developed Based on IRLCOO

The IRLCOO were built internally using the composition pattern that provides a common interface simplifying its use and reusability. The systems use the MVC pattern; the IRLCOO use the View as the external interface of the application. The students interact with the systems through the View dynamically built with IRLCOO. The last versions are based on Flex frameworks; the IRCLOO are composed of prebuilt components, custom components, rich class library, Multimedia eXperience Markup Language (MXML), and AS 3.0. The framework allows users to build RIAs based on FP. The system makes it possible to add an interface based on voice for the educational materials using VoiceXML, enabling a richer interaction with the student. Hibernate is a project that aims to be a complete solution to the problem of managing persistent data in Java. Object/Relational Mapping (ORM) is the name given to automated solutions to the mismatch problem (Peak & Heudecker, 2005). This system implements the persistent data using Hibernate on the server side (https://www.hibernate.org/). The IRLCOO components allow separation of control and content to maximize the reusability.

SiDec 1.0 was used to build Web-based courseware from the stored IRLCOO (Learning Resources). The SiDec lesson templates were based on the cognitive theory of Conceptual Maps

(CM) and Based-Problems Learning (BPL) (Barell, 2006). The SiDeC 2.0 new version supports the IRLCOO new with AS 3.0 and Flex 3.0 to provide RIA online courses. The IRLCOO take learner's metrics, for example, time, learning tracking, and completed activities, with the purpose of tailoring their learning experiences; the metrics are serialized as Semantic Web data using JENA (http://jena.sourceforge.net/). These IRLCOO are compliant with the specifications for learning items of the SCORM 2004 Models (Content Aggregation, Sequencing and Navigation, and Run Time Environment) (http://www.adlnet.gov/Pages/Default.aspx). The IRLCOO use meta data that represent the configuration files at Run-Time, the imsmanifest.xml file describes the control/content to the LMS In our case we use three files imsmanifest.xml to implement the functionality of the manifest SCORM with the purpose to maximize the reusability of the IRLCOO components in the system: the file imsmanifest_container.xml is the configuration file for the container, imsmanifest_content/imsmanifest_evaluation are the configuration files for the content/evaluation and the respective multimedia resources for each, and imsmanifest_navigation is the configuration file for the navigation that is responsible for the course's sequence. The SiDec 2.0 added new lesson templates based on the Case Method (Ellet, 2007), Project-Based Learning (PBL), Uskov (Uskov & Uskov, 2003), and a Structured Open (SO).



```
<IEEE1484LTSA:Content
rdf:ID="urn:rdf:030307ct0000077">
…
<IEEE1484LTSA:Evaluation_name>
Alimentacion_Animales
</IEEE1484LTSA:Evaluation_name>

<IEEE1484LTSA:records rdf:resource="http://148.204.45.65/
rpv/education/EC#urn:rdf:030307et0000033" />

…
</IEEE1484LTSA:Content>
```

```
<IEEE1484LTSA:Evaluation
rdf:ID="urn:rdf:030307et0000033">
…
<Evaluation:title>Historia de los autos</Evaluation:title>
<Evaluation:IRLCOO_type>
FillInTheBlank
</Evaluation:IRLCOO_type>
…
</IEEE1484LTSA:Evaluation>
```

ALIMENTACION DE LOS ANIMALES.

¿SABES CÓMO SE ALIMENTAN LOS ANIMALES?

Los animales, como todos los seres vivos, necesitan alimentarse para vivir (ver video 1: Introducción). Las plantas son organismos autótrofos porque fabrican su propia comida utilizando la luz del Sol (oír audio1: Algas). Los animales, en cambio, son organismos heterótrofos porque no son capaces de fabricar su propia comida y tienen que alimentarse de vegetales y de otros animales (oír audio2: Águila Pescadora).

Se pueden distinguir tres tipos de animales según sea su alimentación: herbívoros, carnívoros y omnívoros.

Los animales que solo comen plantas se llaman herbívoros (oír audio3: Alimento de Herbívoros). Algunos mamíferos, como las jirafas, los elefantes, las cebras, los osos panda, las vacas, los caballos, los conejos, los ciervos o las ovejas, son herbívoros. Las abejas, las mariposas, los murciélagos y los colibríes chupan el néctar de las flores.

Los animales que solo comen carne se llaman carnívoros. Estos se pueden alimentar de animales vivos o de otros que ya están muertos. Muchos carnívoros son depredadores; eso significa que cazan las presas de las que se alimentan. Los leones, los leopardos, los tigres, los lobos, los tiburones, las águilas o las arañas son animales carnívoros. Algunos son muy rápidos y persiguen a sus presas corriendo, como los leopardos y los lobos; otros, sin embargo, prefieren permanecer escondidos o quietos esperando a que alguna presa se

¿En qué año se produjo el primer automovil?

Introduzca su respuesta:

Una carcacha muy bonita!

Figure 4. IRLCOO with meta data.

The Evaluation System 2.0 (ES 2.0) is designed using the same model used for the SiDeC 2.0; the principal difference is that the ES 2.0 performs an analysis of the learner's profile using the MAS and Semantic Web, constructed at Run Time during the teaching/learning process with the learner's experiences. The metrics were collected by IRLCOO and serialized as Semantic Web data using JENA. The purpose of the analysis is to offer a personalized feedback and dynamic reconfiguration of the student's educational materials, doing dynamic modifications to the sequence of the course, personalized to learner's needs, based on the learning process design, according to the obtained results. The ES 2.0 invokes the MAS; the Coach Agent exchanges facts as input and output with the inference engine via the StudentTracker and provides facts to the query engine. The inference engine usees facts and ontologies to derive additional factual knowledge that is implicated, implementing an assistance system based on Web Semantic.

Semantic Web Platform

Tim Berners-Lee has a vision for the future of the Web. The first is to make the Web a more collaborative environment. The second is to make the Web understandable and, thus, processable by machines. Tim Berners-Lee's original vision clearly involved more than retrieving Hyper Text 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 RDF. The key idea is that the original vision encompassed additional 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 (Daconta, Obrst, & Smith, 2003).

The Semantic Web development environment used is the following: Java 1.6 Software Development Kit (SDK) (Oracle, 2009), Jena Semantic Web Framework (http://jena.sourceforge.net/), Protégé Ontology Editor 4.0 Alpha (Protégé 4, 2009), Ontology Reasoner Pellet 1.5.2 (Clark & Parsia, 2009) and Java Agent Development Framework 3.7 (JADE) (http://jade.tilab.com/). The tools use the Java programming language. JENA was selected as the Semantic Web framework. 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 (http://jena.sourceforge.net/).

StudentTracker

The StudentTracker integrates several different sources of information online about the student, for example, time, learning tracking, and completed activities, along the student's session via the IRLCOO on client side. The information is combined into aSemantic Web data model where the StudentTracker can navigate, query, and search across all the sources like a unique model. The objective of StudentTracker is to combine information from several different sources of information online, offering alternatives for tailoring the learning resources at Run-Time. Figure 5 shows the Semantic Web Platform, where the StudentTracker is being executed through the Coach Agent.

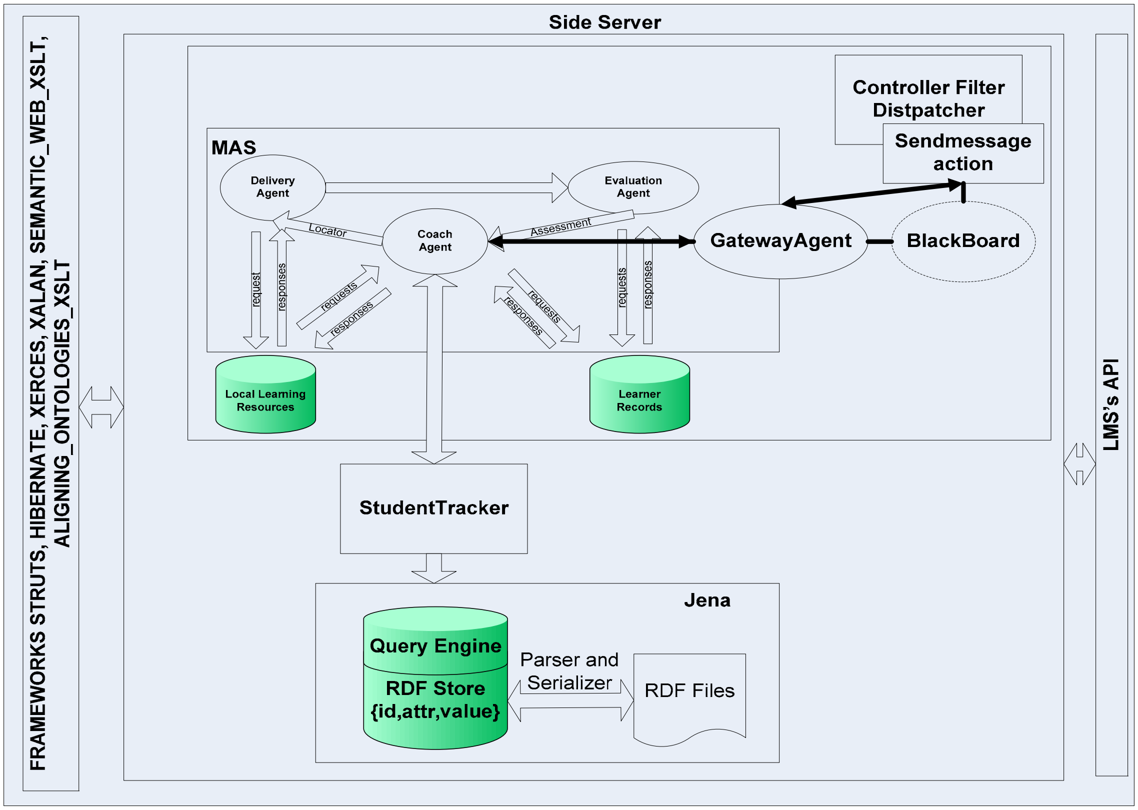


Figure 5. Semantic Web Platform.

The Architecture of the Software Factory

The industrialization of software development has moved the industry towards maturation. The industries customize and assemble standard components to create comparable but different products, to standardize, integrate, and automate production processes, to build tools and configure them to automate recurring tasks, minimizing costs and risks using production lines. Our software factory builds families of contents and evaluations, maximizing the reusability. Assembling IRL-

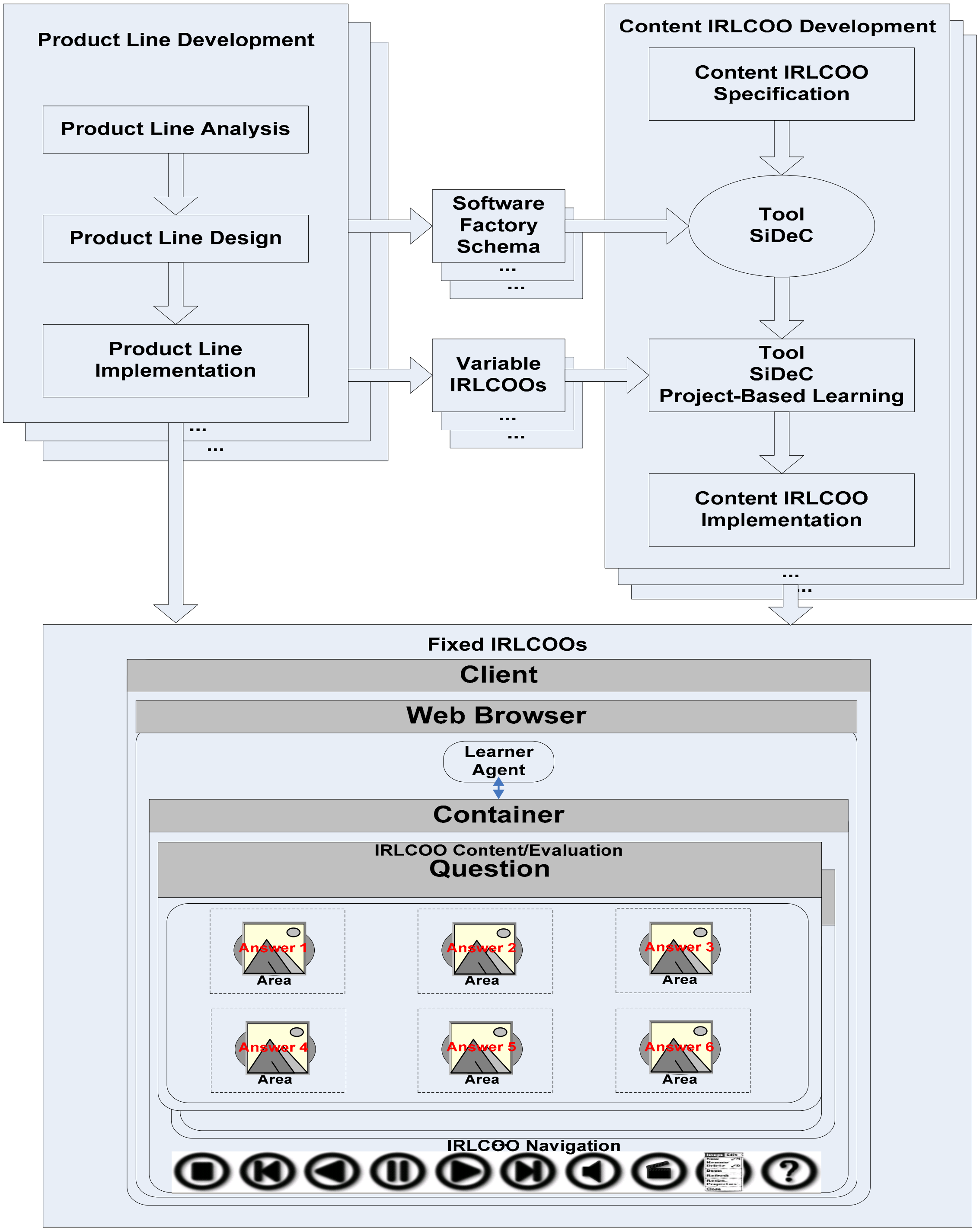


Figure 6. The architecture of the software factory.

COO that consume WS means that the developed tools reduce the building complexity and the amount of handwritten code. Figure 6 shows the architecture of the software factory.

Limitations

The main limitation of the proposed system lies in the components of software based on AS 3.0 and the Flash platform. A component of software is a unit of composition with contractually specified interfaces and explicit context dependencies, the main dependence context is the Virtual Machine to execute code AS 3.0 denominated Flash Player, but enable a cross-platform Run Time Environment which was used to build the components IRLCOO.

Conclusions

The authoring, evaluation, and VoiceXML systems were created under the same architecture based on IRLCOO searching: conquering complexity, managing change, and reusability, keeping in mind the separation of control & content. This architecture is integrated for ACOA, MAS, IRLCOO, and Semantic Web Platform. This approach has focused on reusability, accessibility, durability, and interoperability of the learning contents, which are built as IRLCOO, to be fundamental pieces in the delivery of learning content. Our communication model is composed of the LMS communication API, AJAX, Struts Framework, Hibernate, IRLCOO, WS, Semantic Web, and a server JUDDI. It provides new development capabilities for WBE systems because their integral technologies are complementary. SiDeC 2.0 and the ES 2.0 were developed under this model to help in the automation of and reducing the complexity of learning content development.

The Web Semantic Platform helps to build intelligent and adaptive systems (bidirectional communication) according to the learner's needs. The ADL-Schema manages dynamic sequencing, composition, content separation, and navigation at Run Time Environment (RTE). The proposal has the same ADL advantages and adds the possibility of building Web and desktop applications, using the same learning and evaluation components already generated. The Software Factory schema has automated the development and maintenance of variants of IRLCOO and the production of fixed IRLCOO for adapting, assembling, and configuring ACOA. Finally the architecture, IEEE 1484 LTSA, is improved for collecting certain student's metrics in an automatic way, additionally adding support for the Semantic Web.

Acknowledgments

Authors of this paper would like to thank Superior School of Computer Sciences, Computer Science Research Center - Instituto Politécnico Nacional (IPN), and Universidad Nacional Autónoma de México (UNAM) for the partial support for this project within the project IPN-SIP 20101278.

References

Barell, J. (2006). *Problem-based learning: An inquiry approach*. Thousand Oaks: Corwin Press.

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

Clark & Parsia. (2009). *Pellet: OWL 2 Reasoner for Java*. Retrieved January 7, 2009, from http://clarkparsia.com/pellet/

Daconta, M. C., Obrst, L. J., & Smith, K. T. (2003). *The semantic Web: A guide to the future of XML, Web services, and knowledge management*. Indianapolis: John Wiley & Sons.

Ellet, W. (2007). *The case study handbook: How to read, discuss, and write persuasively about cases*. Boston: Harvard Business Press.

Grane, D., Pascarello, E., & James, D. (2005). *Ajax in action*. Greewich: Manning Publications.

Greenfield, J., Short, K., Cook, S., Kent, S. & Crupi, J. (2004). *Software factories: Assembling applications with patterns, models, frameworks, and tools*. Indianapolis: Wiley.

Gruber, T. (1993). A translation approach to portable ontologies*. Knowledge Acquisition, 5*(2),199-220.

Holmes, J. (2006). *Struts: The complete reference* (2nd ed.). California: McGraw Hill – Osborne Media.

Learn ActionScript. (2009). *Adobe Developer Connection/ActionScript Technology Center*. Retrieved January 10, 2009, from http://www.adobe.com/devnet/actionscript.html

Learning Technology Standards Committee. (2001). *IEEE P1484.1/D9, 2001-11-30 Draft Standard for Learning Technology — Learning Technology Systems Architecture (LTSA)*. Retrieved February 15, 2005, from http://ltsc.ieee.org/wg1/files/IEEE_1484_01_D09_LTSA.pdf

Oracle. (2009). *Java SE Downloads*. Retrieved January 3, 2009, from http://www.oracle.com/technetwork/java/javase/downloads/index.html

Peak, P., & Heudecker, N. (2005). *Hibernate quickly*. Greenwich, CT: 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.

Protégé 4. (2009). *Protégé*. Retrieved January 7, 2009, from http://protege.stanford.edu/download/protege/4.0/installanywhere/

Szyperski, C. (1998). *Component software. Beyond object-oriented programming* (2nd ed.). New York: Addison-Wesley Professional.

Uskov, V., & Uskov, M. (2003). Reusable learning objects approach to Web-based education. *International Journal of Computer and Applications*, *25*(3), 188-197.

Wang, A., & Qian, K. (2005). *Component-oriented programming*. Georgia: John Wiley & Sons.

World Wide Web Consortium. (n.d.). *Latest SOAP versions*. Retrieved January 10, 2010, from http://www.w3.org/TR/soap/

World Wide Web Consortium. (2006). *Extensible Markup Language (XML)*. Retrieved December 10, 2006, from http://www.w3.org/XML/

World Wide Web Consortium. (2010). *Resource Description Framework (RDF)*. Retrieved November 20, 2006, from http://www.w3.org/RDF/

Biographies

Rubén PEREDO ‐ VALDERRAMA received the B. Sc. degree from the Escuela Superior de Ingeniería Mecánica y Eléctrica, ESIME, the M. Sc. degree from the Centro de Investigación en Computación, (CIC), Instituto Politécnico Nacional, IPN. He is currently a candidate to a Ph.D. degree at CIC – IPN. His main research lines are Web-based education, Semantic Web, multi-agent systems and multimedia. In 1999, he joined CIC ‐ IPN, where he worked in the area of Artificial Intelligent (AI). He joined ESCOM ‐ IPN, where now he is working in the area of Computer Systems Engineering. Currently, he is a level ‐ 1 member of the SNI and has several publications in indexed international, national, and institutional journals. He is author of a book chapter Springer and has several other publications and conference proceedings internationally and nationally.

Alejandro CANALES ‐ CRUZ graduated in communication and electronic engineering and obtained his MSc in microelectronic science from the Instituto Politécnico Nacional, Mexico. He received his PhD in computer science from the Centro de Investigación en Computación of the Instituto Politécnico Nacional, Mexico. His main research lines are Web-based education, mobile collaborative learning, artificial intelligence based on multi-agent systems and design of secure software. He has been author, or co-author, of multiple papers published in international journals as well as chapters of scientific books and in peer reviewed conference proceedings.

Iván PEREDO ‐ VALDERRAMA received the B. Sc. degree from the Universidad Autónoma Metropolitana, UAM, the M. Sc. degree from the Centro de Investigación en Computación, (CIC), Instituto Politécnico Nacional, IPN. His main research lines are Web-based education, Semantic Web, multi-agent systems and multimedia. He has been author, or co-author, of multiple papers published in international journals.