Adaptive and intelligent web based education system: towards an integral architecture and framework

Alejandro Canales, Alejandro Peña†, Rubén Peredo, Humberto Sossa, Agustín Gutiérrez
Centro de Investigación en Computación – IPN, WOLNM‡, UPIICSAD, Unidad Profesional ‘Adolfo López Mateos’
Col. Nueva Industrial Vallejo, Del. Gustavo A, Madero, D.F. 07738, Mexico City, México

Abstract

In this paper it is presented our contribution for carrying out adaptive and intelligent Web-based Education Systems (WBES) that take into account the individual student learning requirements, by means of a holistic architecture and Framework for developing WBES. In addition, three basic modules of the proposed WBES are outlined: an Authoring tool, a Semantic Web-based Evaluation, and a Cognitive Maps-based Student Model. As well, it is stated a Service Oriented Architecture (SOA) oriented to deploy reusable, accessible, durable and interoperable services. The approach enhances the Learning Technology Standard Architecture, proposed by IEEE-LTSA (Learning Technology System Architecture) [1], and the Sharable Content Object Reusable Model (SCORM), claimed by Advanced Distributed Learning (ADL) [2].

Keywords: SiDeC; Evaluation System; IRLCOO; MAS; Student Model.

1. Introduction

Student-centered education pursues that sequencing, authoring content, assessment, and evaluation processes meet the learning goals of the students based upon a student model that the WBES outcomes. Also, the content and evaluation repositories must be suitable for the particular requirements of each individual. But at the same time, they have to be flexible and available for being tailored and used by a wide community of developers and students, respectively. It is necessary the development of a WBES architecture that considers a whole diversity of requirements and provides the needed functionalities based on the facilities of the Web.

As well, the software engineering has to consider the particular specifications claimed by the WBES in order to tailor ad-hoc solutions. Furthermore, authoring reusable components for content and evaluation tasks have to be fulfilled automatically as well as possible. The components help us to reduce the complexity, managing change, and reuse [3]. Thus, in order to deliver adaptive and intelligent teaching-learning experiences, it is necessary the anticipation of causal outcomes by student modeling, in addition to the automatic assessment of the behavior of the students.
Wherefore, the purpose of this paper is to show a new architecture for the WBES development based on the IEEE 1484 LTSA specification [1]. In order to achieve this goal, this paper is organized as follows: In section 2, the IEEE 1484 LTSA Overview is presented; whereas in section 3, learner and environment interactions are analyzed. In section 4 the components system layer and the Intelligent Reusable Learning Components Object Oriented (IRLCOO) is described. Afterwards, in section 5 it is sketched the Cognitive Maps-based Student Model. In sections 6 and 7 the SiDeC (authoring content) and Evaluation System are respectively depicted. Besides, these systems are based on IRLCOO, Semantic Web-based and Framework called Multi-Agent System (MAS). Next, in section 8 the resources involved into the SOA, which is in consonance with the trends of the Web technologies are identified. Finally, in section 9 we conclude with some comments and evaluation critics about the proposal, besides of the further work to be done.

2. IEEE 1484 LTSA Overview

The Learning Technology Standards Committee of the IEEE Computer Society has proposed the IEEE 1484 LTSA specification. This Standard describes 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 [1]. This Standard covers a wide range of learning scenarios and is conformed by five layers as shown in Fig. 1.

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 depicted, for this case only 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 implemented. Each one of these layers is described below:

1. Learner and Environment Interactions: Concerns with the learner's acquisition, transfer, exchange, formulation, and discovery of knowledge and information through interaction with the environment.
2. Learner-Related Design Features: Regarding to the expected effect on learners that have to be considered during the design of learning technology systems.
3. System Components: States the component-based architecture, as identified in human-centred and pervasive paradigms.
4. Implementation Perspectives and Priorities: Describes learning technology systems from a variety of perspectives by referencing subsets of the system components layer.
5. Operational Components and Interoperability - code, APIs, protocols: Describes the generic plug and play, interoperable components and interfaces of information technology-based learning architecture, as identified in the stakeholder perspectives.
3. Learner and Environment Interactions

As a result of the learner and environment interactions analysis, the information flows are obtained as sketched in Fig. 2. In resume, there are five basic types of interaction that take place in WBES: learner-content, learner-tutor, learner-learner, learner-university and tutor-university [4] [5].

The learner-content interaction has been the touchstone of education for a long time. Knowledge acquisition happens when the learner interacts with the instructional content, as a process that organizes new information into preexisting cognitive structures. As a result, it is achieved new changes in the consciousness of the student.

Afterwards, the learning is reinforced through the second type of interaction called learner-tutor. This process carries out activities such as: Seeking and offer explanations, analogies and examples, and elaborating, discussing, and applying content. These interactions pursue to accomplish multiple objectives, as: Encourage learner interest, commit the student with the learning process, foster application of content by skill practice and manipulation of information and ideas, organize strategies for testing and evaluating the extent to which learning is taking place, develop alternative pedagogic strategies, and provide support to the students.

However, in order to enable a WBES to move beyond the transmission paradigm and be truly transformative, a third type of interaction is essential: Learner-learner. The interaction among learners can take place with learners relating directly with each other or in-group settings, with or without the presence of tutor in real time. Learner-learner interaction is an extremely powerful way of “helping learners to think out the content that has been presented and to test it in the exchanges with their peers” [4].

The fourth type of interaction is: learner-university. This interaction begins when the students submit their registration for the courses, and goes on when the university requests feedback to the learner about their opinions, complaints, suggestions, and requests about the courses, tutors and the whole educational system.

The fifth type of interaction is done between tutor-university. This relation is established since the moment a tutor is designated to impart, upgrade or develop a course. The interaction continues as a result of the teaching-learning experiences, assessments, evaluations, and feedbacks.

4. System Components Architecture

As a contribution to the former Systems Components layer 3, depicted in Fig. 3, five processes are identified instead of four in the original specification: learner entity, evaluation, coach, delivery process, and University. Also, there are two repositories: Learner records and Learning resources. Furthermore, fourteen information flows among the components are outlined, instead of thirteen.
Layer 3 is normative in this standard but in this case it is necessary to modify it. First of all, the university process is considered because it biases onto educational system in a direct way. Wherefore, the coach process has been divided into two tasks: Coach and virtual coach, because this process has to adapt to the learners’ individual needs. During the learning process some decisions about sequence, quizzes, and activities can be manually made for the coach and others can be automatically fulfilled for the virtual coach.

Briefly, the overall operation has the following form: 1) Negotiation of the learning styles, strategies, and methods between learner and other stakeholders. As a result, they are stated as learning preferences. 2) Learner information assessment. This new operation collects information about the behaviour of the student along the course, e.g., trajectory, spending time, nomadicity, and navigation. This information is stored into learner records via components. 3) Evaluation of the learner based upon multimedia interactions context. 4) Elicitation of assessments and learner information stemmed from the evaluations. 5) Organization of a learner history database regarding to: Keyboard clicks, mouse clicks, voice responses, choices, written responses, and many details more. 6) Query support to the coach in order to review the learner's assessment and learner information, such as: Preferences, past performance history, and future learning objectives. 7) Functionalities for enabling the virtual coach to review the learner’s behaviour and learner information. In addition to the smart and automatic services for dynamic modifications on the course sequence, according to learner’s needs. 8) Seeking learning resources, via query and catalog info, for appropriate learning content demanded by coach and virtual coach. 9) Extraction of locators from the available catalog info by Coach and virtual coach in order to send them to the delivery process, e.g., a lesson plan or references to contents. 10) Delivery process oriented to extract the learning content and the learner information from the learning resources and the learner records respectively. These tasks are based on locators with the goal to transform the learning content to an interactive and adaptive multimedia presentation to the learner. 11) University support to the whole educational system. This new task provides a way for extra information interchange among the university, students and tutors, such as: Answer to doubts and comments for expressing their opinion in general about the courses, tutors, learners and educational system.

4.1. Delivery process

The purpose of the delivery process is to produce learning content and 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. In this work line, there is a special type of labeled materials called: Intelligent Reusable Learning Components Object Oriented (IRLCOO) which were developed by Peredo, Balladares, and Sheremetov [6]. 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.
4.2. System based on components

According to Szyperski 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 [7]. 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.

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 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 and Qian [3] three major goals pursued by Component-Oriented Programming are considered: Conquering complexity, managing change, and reusability.

4.3. IRLCOO profile

With regards to the SCORM terminology, IRLCOO are a special type of Sharable Content Object (SCO) that represent an alternative approach to content development, which is based on the Reusable Learning Object Strategy Definition stated by ADL [2], IRLCOO are self-contained learning components that are organized as learning resources, which are accessed independently. IRLCOO are digital resources that can be reused to support WBES thru: Live, streaming and prerecorded video and audio, a course module, animations, graphics, applications, Web pages, PDF and Office documents, and other pieces devoted to deliver complete experiences. From a pedagogical perspective, each IRLCOO might play a specific role within an instructional design pattern. IRLCOO can be re-assembled to create new courses or sequenced to tailor individual learning paths.

The use of IRLCOO deals with the following key issues: 1) The IRLCOO must be able to communicate with learning management systems (LMS) using a standardized method that does not depend on the system. 2) The sequence system, that usually is a module of the LMS, defines the navigation rules that a learner uses to move between IRLCOO. 3) IRLCOO own a description that enables designers to seek and find the appropriate IRLCOO for the right job. These considerations offer clear benefits, such as: IRLCOO enable mass-customization of learning with more personalized and content ‘just for the learner’, and for authors, there is the opportunity to seek existing IRLCOO within the organization or from external providers in order to reuse them, save time and money.
4.3.1. IRLCOO platform

Flash’s components are used to development IRLCOO due to Flash is an integrator of media and have a powerful programming language denominated ActionScript 2.0 [8]. The 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 (SC) for the learning elements denominated Learning Interactions (LI). The aim is to generate a library multimedia of IRLCOO for WBES with the purpose to separate the content of the control. Thus, the components use different levels of code inside the Flash Player (FP). 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 are tailored IRLCOO. In addition, this development platform owns certain communication functionalities inside the Application Programming Interface with LMS, MAS, and different frameworks, as Hibernate [9], Struts [10], etc.), and dynamic load of Assets in Run-Time.

The IRLCOO were developed through advanced templates, which are enhanced by nonlinear feedback from the users. This platform offers the functionality of dynamic component generation for the auto-configurable labeled materials. The first version of the IRLCOO implements a Bidirectional Associative Memory Neuronal Net for the nonlinear feedback. The feedback is obtained from two types of approaches: 1) Objective and reactive, through an assessment based upon MAS. 2) Subjective and proactive, by means of a Student Model based on Cognitive Maps. The Neuronal Net is composed by two layers, which are identified by two groups of neurons: \( \{a_1, a_2, ..., a_n\} \) and \( \{b_1, b_2, ..., b_n\} \). Where \( n \) is the umpteenth element of the respective vector. The \( n \times p \) generated connections, where \( n \) is pattern length of \( A \) and \( p \) is the number of training pairs of \( B \), conform a correlation matrix represented by \( W \) in (1) [11]:

\[
W = \sum_k^m A_k^T B_k \\
\]

This matrix is generated by means of vectors \( A = \{a_1, a_2, ..., a_n\} \) and \( B = \{b_1, b_2, ..., b_n\} \), where \( A \) is the input vector that can be composed from different factors that a teacher measures, i.e., time, score, trajectory, number of intents; whilst the output vector \( B \) represents the feedback that the learner receive in function of the input vector \( A \), \( m \) is the first pattern pairs.

The second version of the IRLCOO is deployed into MAS supported by Java Agent platform (JADE). The Client tier owns a Snooper Agent devoted to capture the learner info; whilst the Server layer has three agents: Composition, Buffer and Sequence. These agents are oriented to personalize the learning [6].
4.3.2. Communication between IRLCOO, LMS and Frameworks

The common implementation of our model uses an asynchronous Run-Time Environment (RTE). This RTE is a browser-based delivery platform that initiates the communication with content and evaluation objects. As depicted in Fig. 4, a content object and the LMS implement communication adapters in content and evaluation components’ RTE.

In regards to Flash IRLCOO, this adapter is based on the use of a Framework to communicate Flash components with the LMS. With this baseline, the component incorporates the capability to discover and communicate with the communication adapter. Also, the LMS returns the request data, a message identifying an error condition, or as response that the request can not be attended. At run time, the content and evaluation components send state information for storage learning resources [12].

The static sequencing uses a standard sequence engine called: sequencingEngineDynamic.jsp. This service obtains the ID of the last used IRLCOO, and seeks the following IRLCOO to be delivered from the sequence’s RecordSet by means of SQL query. Regarding to the dynamic sequencing, the functionality is carried out by an engine called: sequencingEngineDynamic.jsp. This engine generates SQL queries in a dynamic fashion. Furthermore, it modifies the learner’s trajectory according to variables from the learner records and from the support of the Sequence Agent.

5. Student Model

The Student Model deals with the conceptualization and maintenance of a mental model that the system achieves about the student along the teaching-learning experiences. This process faces subjective concepts, partial beliefs, uncertainly assumptions and fuzzy criteria. In addition, the Student Modeling can be stated as: The problem of representing, explicitly and faithfully, all aspects of the individual that concern his/her learning [13]. The Student Model must be tailored to each specific student, and it has to be updated permanently to depict the acquisition of knowledge about the teaching domain. In addition, changes in attitude and cognitive performance of the student are included. In resume, Student Modeling could be considered as an extremely difficult problem, according to the study done by Self entitled: Bypassing the intractable problem of student modeling [14].

However, the Student Model arena offers a wide spectrum of targets that focus on issues like: Multiple intelligence [15], affective and motivational mental states [16] [17], learning by reflection [18] [19], participative Student Modeling [20] [13], overlay knowledge acquired by tutor against student [21], perturbation knowledge about student’s misconceptions [22], episodic regarding to student’s behavior [23], cooperative-collaborative [24] [25] [26], negotiated [27]. In spite of these efforts, there is a lack of a fuzzy, qualitative and causal-based approach oriented to outline the factors that stimulate and inhibit the learning of the student, and how they are involved to outcome a particular behavior.
5.1 Student Model Profile

The problem that the Student Model focus is: How to state the factors and estimate their causal effects that bias the learning of the student about a topic, before delivering a teaching-learning experience? Thus, the Student Model tries to sketch and simulate the causal phenomenon stemmed from a teaching-learning experience delivered by the system. The assumption lies on: As a result of measuring the factors involved into the teaching process, outline their relations, and trace causal-inference paths; it is possible to predict the impact on the student’s learning.

Our Student Model is based upon a fuzzy and causal version of a Cognitive Map. Essentially, a Cognitive Map is a tool that can be used for anticipating the causal effects that a teaching-learning experience produces in the knowledge acquisition of the student. The Cognitive Map represents factors elicited from two types of domains: Teaching, and Student. The teaching domain corresponds to concepts that describe content objects, sequencing strategies, and evaluation components. 2) The student domain identifies concepts about his/her personality, learning preferences, and cognitive skills. The relations between the concepts are analyzed from the causal perspective, where there is a cause concept that biases on an effect concept in some way.

The prediction is done by the simulation of the fuzzy-causal influence that teaching domain factors produce on the student, according to his/her domain. Once the teaching-learning experience is delivered to the student, its outcomes are compared against the predictions in order to identify the accuracy of the Student Model.

5.2 Baseline of the Student Model

The conceptual Framework of the Student Model lies on five principles of the Activity Theory as stated by Leont’ev [28], the Student Model Foundations pointed out by Self [29] and Koch [30], and the Rule-Base Fuzzy Cognitive Maps model proposed by Carvalho [31].

The Student Model conceives the teaching-learning phenomenon as the externalization-internalization principle of the Activity Theory. Thus, they are modeled as cause-effect entities in a Cognitive Map. As a cause, the teaching is the externalization process where the individual socializes with the environment surrounding in order to face the teaching experience. As an effect, the internalization process is triggered as the learning activity oriented to acquire knowledge. These experiences are organized into the principle of hierarchy structure that is object-orientedness. This principle claims that every activity pursues an object like the knowledge acquisition. The structure is a taxonomy of activities oriented to objects, which are organized in a set of actions that focus on specific goals. Every action is carried out by a set of operations that are constrained by conditions. The performance of the activity is promoted by the tool principle, which aims to stimulating the activity of the individual trough the support of an agent. Thus, in this context the system as a whole and more particularly the Student Model play the role tool. Finally, the principle of development is accomplished by the outcomes derived from the teaching-learning experiences that the student faces, against the predictions done by the Student Model. In
sort, the Activity Theory pursues the anticipatory reflection before developing any action. This aim is in consonance with the orientation of the Student Model proposed.

The underlying concepts of the Student Model’s are based upon the beliefs Framework stated by Self [29] and Koch [30], that rest on the knowledge and assumptions that the system has about the student. The beliefs \( \mathcal{B} \) are represented by propositions \( p \) from the propositional calculus, which may be true or false according to the perspective of the system \( s \) or the student \( u \). So the set of propositions that the system believes \( \mathcal{B}_s \) is depicted in (1); where \( \mathcal{B}_{sp} \) is the set of propositions that can be nested. Thus, the set of propositions \( p \) the system \( s \) believes \( \mathcal{B}_s \) are believed by the student \( \mathcal{B}_{up} \) is defined as \( \mathcal{B}_{su} \) in (2). However as the system unknowns the user’s beliefs, all reasoning has to be done based upon the system’s beliefs. Therefore the subset of beliefs that the system owns about the student is the student model \( \mathcal{USM} \) are defined as:

\[
\mathcal{B}_s = \{ p \mid \mathcal{B}_{sp} \} \quad \text{(2)}
\]

\[
\mathcal{B}_{su} = \{ p \mid \mathcal{B}_s \} \quad \text{(3)}
\]

\[
\mathcal{USM} = \mathcal{B}_s \cup \mathcal{B}_{up} \quad \text{(4)}
\]

The beliefs \( \mathcal{B} \) could be replaced with knowledge \( \mathcal{K} \). So a system \( s \) knows \( p \) if \( \mathcal{K}_{sp} = \mathcal{B}_{sp} \), and \( p \) is true and there is a source for knowing that \( p \) is true. With this baseline, it is possible to define a Student Model. Wherefore, in the Student Model five types of domains are considered, such as: Knowledge Teaching \( \mathcal{T} \), Cognitive \( \mathcal{C} \), Personality \( \mathcal{P} \), Learning Preferences \( \mathcal{L} \), and Independent Knowledge \( \mathcal{I} \); which are depicted in equation (5) to (9), as the set of propositions related to the particular domain \( \mathcal{T}, \mathcal{C}, \mathcal{P}, \mathcal{L}, \mathcal{I} \) that the system \( s \) believes and the set of propositions related to the specific domain. In sort, these models lead to depict the whole description about the Student Model \( \mathcal{UM} \) as is shown in (10).

\[
\mathcal{T}_s (U) = \{ p \mid \mathcal{B}_{sp} \cap p \in T \} \quad \text{(5)}
\]

\[
\mathcal{C}_s (U) = \{ p \mid \mathcal{B}_{sp} \cap p \in C \} \quad \text{(6)}
\]

\[
\mathcal{P}_s (U) = \{ p \mid \mathcal{B}_{sp} \cap p \in P \} \quad \text{(7)}
\]

\[
\mathcal{L}_s (U) = \{ p \mid \mathcal{B}_{sp} \cap p \in L \} \quad \text{(8)}
\]

\[
\mathcal{I}_s (U) = \{ p \mid \mathcal{B}_{sp} \cap p \in I \} \quad \text{(9)}
\]

\[
\mathcal{UM} = \mathcal{T}_s (U) \cup \mathcal{C}_s (U) \cup \mathcal{P}_s (U) \cup \mathcal{L}_s (U) \cup \mathcal{I}_s (U) \quad \text{(10)}
\]

This Cognitive Maps-based Student Model \( \mathcal{CMbSM} \) is based on the philosophy principle about causality that states: For every fact there is at least one cause, and given the same conditions, the same causes bias the same effects [31]. The causality is a kind of relationship between events that is able to produce the occurrence of one of them. The events are associated to abstract or physical entities that are pointed out as concepts.

So the concepts outline the main attributes of the entities and identify their internal relationships. A causal relation is a belief about the way a concept promotes or inhibits another. The causal influence among the concepts is sketched by a digraph as depicted in Fig. 5; where there are four nodes regarding to the concepts: Negative, creative, towards action, and algorithms. These factors correspond to the personality, cognitive, learning preferences, and knowledge domains respectively. In this Cognitive Map the links represent the causal relations between concepts.
Each causal relation is described by means a Rule Base-Fuzzy Causal Relation (RB_FCR), which has an id-number and a sign that depicts a positive (+) or negative (-) bias. A typical content of a RB_FCR is instanced in Table 1, where it is outlined the fuzzy-causal relationship between the attitude negative cause concept and the preference toward action effect concept. In the left column of Table 1, appear several instances from the universe of discourse of the antecedent fuzzy value assigned to the cause concept, whilst in the right side of Table 1, it is identified the corresponding causal value in the universe of discourse of the consequent-effect concept.

The outcome of the rules in a RB_FCR does not represent the state value of the effect concept, but it expresses a kind of perturbation about the state of the concept, as a fuzzy variation. The fuzzy-causal inference is achieved through simulation. Prior start the cycle, the fuzzy values of the concepts are measured and introduced as initial values. In Fig. 5, the fuzzy values are identified: Increase, maintains, decrease slow, and improves for negative, creative, towards action, and algorithms respectively. At each step, the causal effects that simultaneously bias the concepts are accumulated according to the causal paths sketched in the Cognitive Map of Fig. 5.

The new concepts’ fuzzy values are computed over the universe of discourse associated to the concept by means of a Fuzzy Carry Accumulation [31]. So the fuzzy membership values, which correspond to the activity intensity in the universe of discourse, of each concept are estimated along the iterations to depict a histogram like the shown in Fig. 6.

The process ends when the concepts’ values reach an equilibrium state, a limit cycle composed by several values patterns, or a chaotic attractor without fixed patterns. For example, in Fig. 7, the four concepts achieve a stability value after \( n \) iterations due to there is no more different values. The behavior of the concepts and the whole Cognitive Map is revealed from the variation of the concepts’ values along the iterations, and from the final state that they reach. As a result of this interpretation, the prediction about the causal outcome is done.

In sort, the domain concepts involved in the Cognitive Map reveal the beliefs that the system has about the student. In this Student Model, it is considered that each teaching-learning experience is authored in several options of Content type, Sequencing strategy, Assessment method, and Evaluation mechanism. The knowledge domain is depicted as concepts, whose initial values represent the background of the student about the subject. Thus, these concepts values are progressively perturbed as a result of the teaching-learning experiences faced by the student.

### 5.3 Student Model Architecture

The Student Model process is organized in three stages: Development, Initialization and Exploitation. The Development stage is accomplished by five modules: 1) Neural nets-based pattern recognition profile about cognitive, personality and learning preferences of the student. 2) Tool for authoring teaching-learning experiences. 3) Ontology administrator about the concepts of the context domain. 4) Evolutionary mechanism that automatically generates Cognitive Maps. 5) Fuzzy causal inference engine.
The Initialization stage is fulfilled by three processes: 1) Student Profile elicitation, as a result of the psychological test that the student faces. 2) Education Program Profile, as a consequence of the description of the sequencing, assessment, content, and evaluation strategies used by the system to deliver teaching-learning experiences. 3) Ontology acquisition, all attributes for each option of teaching-learning experiences are identified in the ontology about the domain.

The third state, Exploitation, corresponds to the teaching-learning cycle. At each iteration the sequencing task, content delivering, assessment record and evaluation task are done. The selection of the option of the teaching-learning experience to be delivered is based on the simulation results provided by the corresponding Cognitive Map. The concepts involved in the Cognitive Map represent the student profile and the characteristics of the experience’s option.

The Student Model plays two roles as a tool for planning and controlling the teaching process. With regard to first role, the Student Model provides subjective beliefs about the causal effect that the components to be delivered by the content and evaluation systems could produce on the student. Thus, its role is proactive, trying to anticipate the learning achievements before delivering any option of teaching-learning experience.

Once the teaching-learning experience has been delivered, the Student Model performs its second role as a control object that is updated. The tuning applied to the Student Model is done according to the accuracy of its predictions. Thus, the outcomes and information stemmed from the content and evaluation systems are used against the predictions for identifying deviations and misconceptions.

6. SiDeC

In order to facilitate the development of learning content, it was built an authoring tool called eCourses Development System (SiDeC). SiDeC is a tool based on components, which facilitates the authoring content by 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 in the way illustrated in Figs. 8 and 9. 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 [32], but in the future we will consider others theories such as: Based-problems learning (BPL), the cases method, and the project method.

The inclusion of cognitive elements, as the Conceptual Maps, obeys to the instructional design pattern for the courses development. Thus, the courses do not only have theoretical or practical questions, but rather they include a mental model about individual thought process. Conceptual Maps are a schema to structure concepts with the purpose of helping the learners to maximize the knowledge acquisition. A Conceptual Map 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 his/her 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. Metadata represent the specific description of the component and its contents, such as: Title, description, keywords, learning objectives, item type, and use rights. The metadata tool provides templates for entering metadata and storing each component in the SiDeC or another IMS/IEEE standard repository.

The specifications of the IRLCOO are encoded into extended markup language (XML [35]). The specifications are deliverables of the tool, as a result of label and structure didactic materials. XML is considered as a specification language for components, and a graph coding that enables generated components being compatible with EDUCOM /IMS compliant RTE. For illustrative purposes of this Framework, in Fig. 10 is sketched the use of IRLCOO within the Web navigator at Run-Time.

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). These atoms are grouped together to teach a common task based on a single learning objective, as is depicted in Fig. 11 [33]. A RLA is an elementary non-dividable piece of learning that is built upon a single learning objective. Each RLA can be classified as: Concept, fact, process, or procedure. Each RLA combines three types of component objects, specifically: 1) Content components, which are pieces of learning materials that are structured based on pedagogical forms, as conceptual maps. 2) Practice components, that represent learning activities. 3) Assessment components, that are organized as test, quizzes, and exams. 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.

On the other hand, a RIA is an elementary no divisible 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. The summary RIA may contain several content items such as: Review, next step, and references/links/additional knowledge resources. These components are used in order to convert the collection of components into a complete ‘learning experience’. As well, there are four types of RIA’s more, they are: Welcome RIA, Overview or Introduction RIA, and Summary RIA. These information atoms are developed by IRLCOO, and they are added to a set of RLAs.

In Fig. 11, the SiDeC implements the conceptual map as a navigation map, allowing to the learner interacts with content objects along the learning experiences. These experiences follow an instructional-teaching strategy. There kind of strategies carry out modifications of the learning content structure. Such modifications are done by the learning experience designer with the objective of provide significant learning, and to teach the learners how to think [34].

Based on a conceptual map the SiDeC represents the course structure that the student follows. The delivery process identifies a learning content for the student. The learning content owns IRLCOO associated with it. Afterwards, the delivery process launches the
IRLCOO and presents them to the student. Fig. 12 depicts how the course structure is organized as a manifest, and the learning content can be interpreted in a Learning Content Tree. A Learning Content Tree is a conceptual structure of learning activities managed by the delivery process for each learner. The Tree representation is just a different way for presenting content structure and navigation. This information is found in the manifest that is defined into the SCORM Content Aggregation Model (CAM).

The SCORM CAM describes the IRLCOO used in a learning experience, and how to package those components for exchange between systems. Furthermore, CAM depicts how those components are available for searching and discovery. As well, CAM defines the sequencing rules for the components, and promotes consistent storage. In addition, CAM aims for labeling content thru meta-data, besides of packaging, exchange and discovery of content. The SCORM CAM also defines responsibilities and requirements for building content aggregations, e.g., course, lessons, modules.

With regards to the coach/virtual coach process, this evaluates the learner performance and navigation requests in order to change the course sequence if it is necessary. The baseline of this process corresponds to the SCORM Sequencing and Navigation Model. This model defines the process sequencing and navigation in terms of a Learning Content Tree, which is a conceptual structure of learning activities for each student managed by the delivery process.

The sequencing of experiences is based on two underlying sources: Student Model and Assessment. From the Student Model are stemmed predictive outcomes; whilst from the Assessment are interpreted real facts about the student’s behavior. The predictions provided by the Student Model come from the simulation achieved through cognitive maps, whilst the feedback assessment is fulfilled by the MAS environment. In addition, the sequence defined for the instructional strategy is supported by Conceptual Maps.

The SCORM Run Time Environment Model describes how identified IRLCOO are launched at run time. The sequence for launching IRLCOO for a given learner is based on a content structure. This structure provides a learning experience through the student interaction with IRLCOO. The SCORM Run Time Environment Model describes how the delivery process manages the resulting learning experience and how that learning experience may affect the Learning Content Tree.

7. Evaluation System

The Evaluation System of our adaptive and intelligent WBES 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 student’s behavior at Run-Time. These measures are stored into learner records that compose the profile. The generation of new sequences of courses is in function of the obtained results, besides the account of the adaptation level.

The Evaluation System combines IRLCOO, additional meta labels, and a Java Agent platform. Also, some technologies of the Intelligence Artificial 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 the potential developers, standardization of functionalities and attributes, definition of a set of specialized APIs, and deployment of a Semantic Web platform. The Semantic Web’s architecture is composed by the nine layers showed in Fig. 13.

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 [35]. The Resource Description Framework (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 [36] [37]. The Ontology tier is devoted to define the semantic for establishing the usage of words and terms in the context of the vocabulary [38]. Logical level corresponds to the reasoning used to establish consistency and correctness of data sets and to infer conclusions that are not explicitly stated [39]. 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 [40].

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 made by content IRLCOO. 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 analyses the learner's records carried out by the MAS. This agent platform owns four agents: Snooper, Client, Buffer, Composition and Sequence. The fundamental idea is to automate the learner's analysis for the coach/virtual coach, and to give partial results that can be useful for the learner's final evaluation. These agents are implemented as JavaBeans programs, which are embedded in the applications running both at the client and server sides. The agent Snooper works as a trigger by means of the INFORM performative that activates the server’s part of the MAS. This agent is deployed into a Java Server Page that uses a JavaBean. Once the evaluation is finished, the graphical user interface activates the Snooper agent and sends the evaluation metrics to be analyzed at the server-side of the MAS. The Snooper agent activates the Buffer agent, which stores the evaluation results in the learner records. These results are represented as a string dynamically constructed by the rule-based inference engine known as JENA, which is based on the evaluation parameters. The Composition and Sequence agents take this information and transform it into a modified composition and sequence of the resources for the student. These functionalities take into account the metrics measures and the logics established by the coach. In resume, the components and operation of the Evaluation System is outlined in Fig. 14.
8. System Open Architecture

The adaptive and intelligent WBES proposed in this work is supported by a set of facilities, resources and technologies that are in consonance with the trends of the Web research. In addition, the aim is to provide services and contents that are reusable, accessible, durable and interoperable. Wherefore, the platform that integrates the resources, tools and facilities is called: Services-Oriented Architecture (SOA). The SOA is sketched in Fig. 15, where each application is registered as a federative service that is available to the whole Web community without constraints of compliance.

For instance, in relation to the Student Model based on Cognitive Maps, the facilities used for its deployment corresponds to the Microsoft® Dot Net Framework®. The programs are encoded with C#® language, which is an Object Oriented Language that is a member of the Visual Studio 2005® suite. Only the client-tier is a Web form, and the remaining applications, middle and back tiers, are agents that operate as Web Services.

The SiDeC and the Evaluation System were developed under the platforms Java 2 Enterprise Edition (J2EE) and Dot Net Framework®. The code follows the guidelines of the Components-Oriented Programming (Java and C#®). These systems use an open middleware called MAS, where the agents are deployed as Web Services. In addition, others technologies are incorporated, such as: Flash, JADE and JENA. The Data Base engine is Microsoft SQL Server®.

The access to the whole systems and their services is possible through SOA, which is based upon the second generation of Web Services. SOA’s architecture has five layers of protocols, languages and services as: 1) Hypertext Transfer Protocol (HTTP). 2) XML. 3) Simple Object Access Protocol (SOAP). 4) Web Services Description Language (WSDL). 5) Universal Description, Discovery and Integration (UDDI).

9. Discussion and Conclusion

This work has introduced an instance of an adaptive and intelligent WBES. Our approach focus on: reusability, accessibility, durability, and, interoperability of the learning contents, which are built as IRLCOO, as the main component for delivering teaching and evaluation content.

IRLCOO offer a common interface and functionality that makes easy the authoring of learning content that is delivered by dynamic sequencing. The IRLCOO accept feedback via assessment based upon MAS platform. The information provided is considered as a rough data because it is based on parameters elicited from the behavior of the student.

Furthermore, the predictions stemmed from the Student Model are dealt as subjective input because it is the result of causal inference over the teaching and students domains factors. The Student Model sketches by means of cause-effect relationships the externalization-internalization cognitive process that happens in the teaching-learning experiences. However, this approach lacks the consideration of issues regarding to how the cognitive process is achieved in the mind of the student, or how the individual stores the knowledge acquired.
A distributed model of the knowledge space composed of IRLCOO integrates different topics into the common Framework. The development of this model is supported by the SiDeC and the Evaluation System. Regarding to the specification and the packing of learning content, the work is done under the SCORM 2004 standard. The use of these systems allows reducing drastically the development time for authoring learning content.

The agent platform MAS is used as an open middleware implementation. The technical goal of MAS is clear: To deploy distributed information technologies so that the availability of services may be more efficient and flexible. Among the advantages stemmed from the use of MAS are: Systems integration can be performed in a high degree, the functionality of the IRLCOO is increased substantially; as well the availability for implementing different techniques, learning styles, instructional strategies, and interaction techniques.

Nowadays, we are also working on smart agents that use XML as the communication language and work with the second generation of Web Services. In addition, we are working out on learning styles that are suitable to apply to the WBES.

As a further work is the evaluation of the Student Model according to the analysis of variance (ANOVA) and covariance (ANCOVA). Particularly, the accuracy of the predictions given by the Student Model is based upon the independent, dependent and nuisance variables. Also, the measure of the effect size and the power, which correspond respectively to the treatment magnitude and the inverse sensitivity is considered [41].

Acknowledgments

Authors of this paper would like to thank the Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC) and Consejo Nacional de Ciencia y Tecnología (CONACYT) for the partial support for this project within the projects CONACYT 143049, CONACYT SEP-2004-C01-46805/747, IPN 20060064. Whilst, A. Peña gives testimony of the strength given by his Father, brother Jesus and Helper, as part of the research projects of World Outreach Light to the Nations Ministries (WOLNM), with the assistance of CONACYT 182329, IPN: SIP-20060064, COFAA DEBEC/138/06, COTEPABE PL-430/05, & Microsoft México.

References

technology to maximize learning and meet business objectives. San Francisco: Jossey-Bass. USA; (pp. 19-36).


IEEE 1484.2, Draft Standard for ECMAScript API for content to runtime services communication.

Fig. 1. IEEE 1484 LTSA (2001) specification.

Fig. 2. Learner entity.

Fig. 3. Layer 3 Systems Components.
Fig. 4. Learning content ECMAScript API model (adopted from IEEE).

Fig. 5. Cognitive Map, about designing algorithms
<table>
<thead>
<tr>
<th>Antecedent: Attitude</th>
<th>Consequent: Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Negative</strong></td>
<td><strong>Toward Action</strong></td>
</tr>
<tr>
<td>If decrease very high</td>
<td>Then stimulated high</td>
</tr>
<tr>
<td>If decrease high</td>
<td>Then stimulated high</td>
</tr>
<tr>
<td>If decrease low</td>
<td>Then maintains</td>
</tr>
<tr>
<td>If maintains</td>
<td>Then increase low</td>
</tr>
<tr>
<td>If increase</td>
<td>Then increase low</td>
</tr>
<tr>
<td>If increase low</td>
<td>Then increase high</td>
</tr>
<tr>
<td>If increase high</td>
<td>Then increase high</td>
</tr>
</tbody>
</table>

Table 1. Rule Base Fuzzy Causal Relation (RB_FCR 5-) between two concepts

![Concepts Histogram](image)

Fig. 6. Concepts Histogram that illustrate the evolution of the Cognitive Map.
Fig. 7. Stability values achieved by the concepts of the Cognitive Map

Fig. 8. SiDeC general operation from an initial perspective.

Fig. 9. Learning content generated for the SiDeC.
Fig. 10. IRLCOO construction scheme in Run-Time.

Fig. 11. Learning content generated for the SiDeC.
Fig. 12. Course structure (CM) generated by the SiDeC based on SCORM CAM/SN/RTE.

Fig. 13. A layered approach to the Semantic Web.
Fig. 14. Evaluation System Architecture using Semantic Web and Components.

Fig. 15. SOA for WBES.