Sequencing of Learning Objects based on SCORM Using cmi Elements and JavaScript

Miguel Sánchez-Brito[1], José Ruiz-Ascencio[1], Carlos Felipe García-Hernández[2]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Cuernavaca, Mexico

[2] Instituto de Investigaciones Eléctricas (IIE), Cuernavaca, Mexico

{miguelsb, josera}@cenidet.edu.mx, cfgarcia@iie.org.mx
http://www.cenidet.edu.mx

Abstract. In this research work a method for sequencing learning objects based on the SCORM standard using cmi tracking elements from this standard and javascript programming language, is presented. Advanced sequencing rules, dependent on student performance are achieved with SCORM 1.2, surpassing the present capabilities of learning objects in learning management systems. An analysis of a case study is also presented, showing that the configured code works correctly.

Keywords: SCORM, Moodle, cmi, learning object, LO, Captivate, JavaScript, sequencing, SCO, Multi SCORM, LMS, ADL, Reload Editor, on-line course.

1 Introduction

At present, more and more educational institutions have increased their attendance capacity, thanks to on-line education, that allows a student to carry out his/her school activities from any place with a computer connected to Internet, at any time. When choosing self-education, in most cases, on-line education requires Learning Objects (LO) with a proper pedagogical support, and attractiveness to students. Additionally, it is of great importance that the teaching material be adaptable to each student, based on his/her interactions in a specific activity of the teaching material. In other words, the learning process will be improved when using an LO that is really a challenge for the student. In this research work, a method for sequencing LOs based on "JavaScript" programming language, is presented, including the cmi tracking elements of SCORM standard (Shareable Content Object Reference Model, version 1.2), Captivate authoring tool from Adobe and the Learning Management System (LMS) Moodle (version 1.9.+) [4].

This paper is organized as follow: Section 2 provides related work on Shareable Content Object (SCO) sequencing and an application of pedagogical tools to an on-line course for e-learning. Section 3 presents the LO, SCORM standard, cmi and their

pp. 9–16; rec. 2015-08-28; acc. 2015-10-02

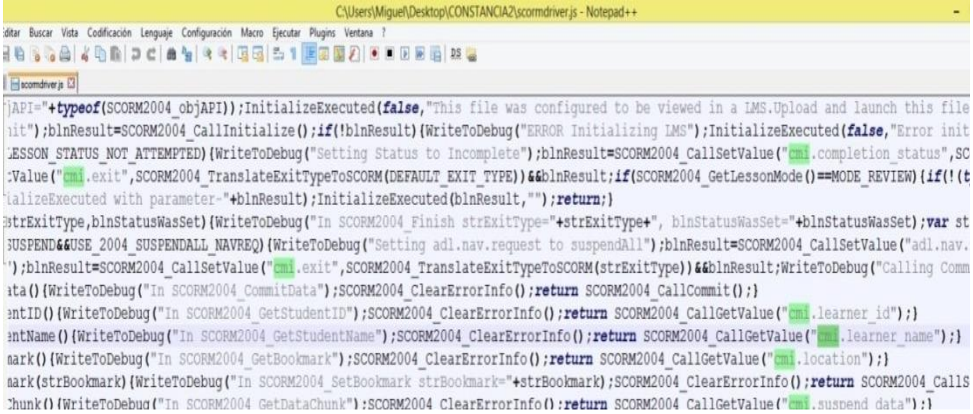
capabilities in Moodle. Section 4 covers the capture of tracking elements in the LMS´s LO. Section 5 presents how to get the cmi values using javascript. Section 6 presents the resulting performance evaluation of code. Finally Section 7 and 8 cover the future work and conclusion for this research.

2 Related Work

In García-Hernández et al. authors explain in detail the way to achieve configuration of SCORM activities [5], involving sub-activities or several Shared Content Objects (SCO as named in SCORM standard, or learning objects containing teaching materials), following the guidelines proposed by ADL with Template 10 and Educative Model 3, and using Reload Editor software. Two forms of achieving sub-activities sequencing of a SCORM activity are described: applying learning objectives mapping for each activity (with the authoring tool) and setting ADL objectives with Reload in order to evaluate them, using shared global objectives and cumulative sub-activities (roll out). In Anbar et al. [6] authors explain an application of pedagogical tools to an on-line course for e-learning, including an example of a script for an activity, and a course development model.

3 LO, SCORM Standard, cmi and their Capabilities in Moodle

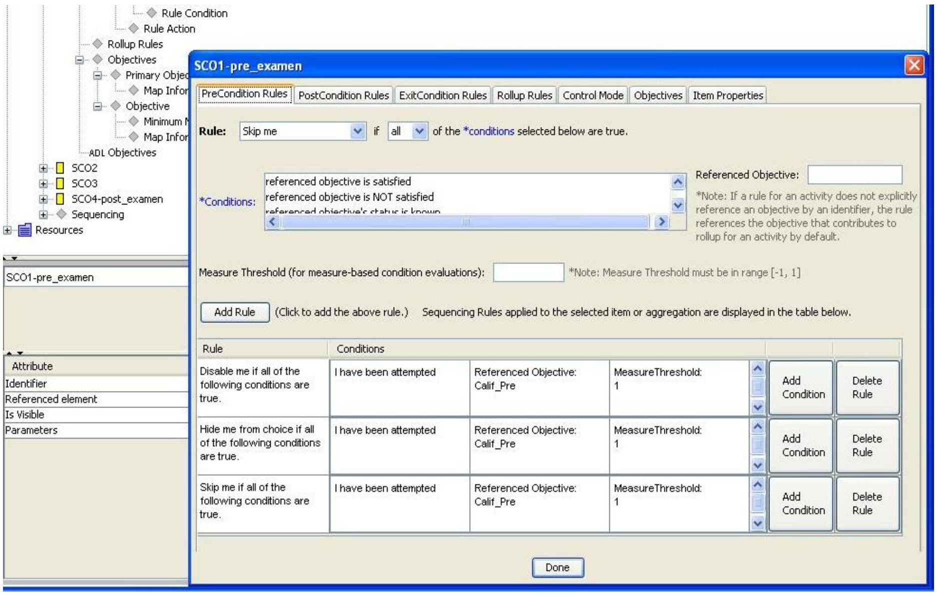
An LO is a teaching block configured with different multimedia elements (text, audio, images, animations, videos, etc.), aimed to motivate the learning process in a specific subject. Advanced Distributed Learning (ADL), a U.S.A. organization, is in charge of updating and publishing the SCORM Standard, the most recent version being the “2004 Fourth Edition”. SCORM is a standard that indicates how to develop teaching material (LO), for on-line learning (e-Learning) [2].


C:\Users\Miguel\Desktop\CONSTANCIA2\scormdriver.js - Notepad++
ditar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
scormdriver.js

```
jAPI="+typeof(SCORM2004_objAPI));InitializeExecuted(false,"This file was configured to be viewed in a LMS.Upload and launch this file
iit");blnResult=SCORM2004_CallInitialize();if(!blnResult){WriteToDebug("ERROR Initializing LMS");InitializeExecuted(false,"Error init
LESSON_STATUS_NOT_ATTEMPTED){WriteToDebug("Setting Status to Incomplete");blnResult=SCORM2004_CallSetValue("cmi.completion_status",SC
:Value("cmi.exit",SCORM2004_TranslateExitTypeToSCORM(DEFAULT_EXIT_TYPE))&&blnResult;if(SCORM2004_GetLessonMode()==MODE_REVIEW){if(!(t
ializeExecuted with parameter-"+blnResult);InitializeExecuted(blnResult,"");return;}
strExitType,blnStatusWasSet){WriteToDebug("In SCORM2004_Finish strExitType="+strExitType+", blnStatusWasSet="+blnStatusWasSet);var st
SUSPEND&&USE_2004_SUSPENDALL_NAVREQ){WriteToDebug("Setting adl.nav.request to suspendAll");blnResult=SCORM2004_CallSetValue("adl.nav.
");blnResult=SCORM2004_CallSetValue("cmi.exit",SCORM2004_TranslateExitTypeToSCORM(strExitType))&&blnResult;WriteToDebug("Calling Comm
ata(){WriteToDebug("In SCORM2004_CommitData");SCORM2004_ClearErrorInfo();return SCORM2004_CallCommit();}
entID(){WriteToDebug("In SCORM2004_GetStudentID");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.learner_id");}
entName(){WriteToDebug("In SCORM2004_GetStudentName");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.learner_name");}
mark(){WriteToDebug("In SCORM2004_GetBookmark");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.location");}
mark(strBookmark){WriteToDebug("In SCORM2004_SetBookmark strBookmark="+strBookmark);SCORM2004_ClearErrorInfo();return SCORM2004_CallS
Chunk(){WriteToDebug("In SCORM2004_GetDataChunk");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.suspend_data");}
```

Fig.1. cmi elements of an LO.

Development of an LO based on SCORM is carried out with an authoring tool (Captivate, for this case study). Captivate is for editing text, inserting images, videos, animations, etc., and once the configuration of an LO is finished, it is published ("Save As" .zip extension file), in order to install it in LMS platforms. When publishing an LO, several code files are generated, that allow communication between an LO and an LMS platform. When using Captivate, a file called “scormdriver.js” is generated (among other files), which contains all the tracking elements or cmi elements of the LO. In figure 1, some cmi elements are shown.



SCO1-pre_examen

Attribute
Identifier
Referenced element
Is Visible
Parameters

Rule: Skip me if all of the *conditions selected below are true.

*Conditions:
referenced objective is satisfied
referenced objective is NOT satisfied
referenced objective's status is known

Referenced Objective:

*Note: If a rule for an activity does not explicitly reference an objective by an identifier, the rule references the objective that contributes to rollup for an activity by default.

Measure Threshold (for measure-based condition evaluations):

*Note: Measure Threshold must be in range [-1, 1]

(Click to add the above rule.) Sequencing Rules applied to the selected item or aggregation are displayed in the table below.

Rule	Conditions				
Disable me if all of the following conditions are true.	I have been attempted	Referenced Objective: Calif_Pre	MeasureThreshold: 1	Add Condition	Delete Rule
Hide me from choice if all of the following conditions are true.	I have been attempted	Referenced Objective: Calif_Pre	MeasureThreshold: 1	Add Condition	Delete Rule
Skip me if all of the following conditions are true.	I have been attempted	Referenced Objective: Calif_Pre	MeasureThreshold: 1	Add Condition	Delete Rule

Fig.2. Configuration of advanced sequencing rules with ReLoad Editor is presented.

An LMS is a software system that carries out several activities required in an on-line school (user enrollment, activities, exams, internet links, students follow-up, grade book, etc.).One of the most relevant activities for the learning process, is the LO. However, at present most LMS (including Moodle) do not work correctly with an LO based on SCORM 2004, since they only support version 1.2 (previous to version 2004). Hence, this presents a great inconvenience for the learning process, because SCORM 2004´s capabilities [1, 3] allow presenting LOs with more than one activity or Multi SCORM (M-SCO), according to this standard. They also allow presenting it in an organized way that depends on objectives evaluation, completeness state, progress percentage and LO revision time. Hence, achieving advanced sequencing rules with SCORM 1.2 is the goal of this work.

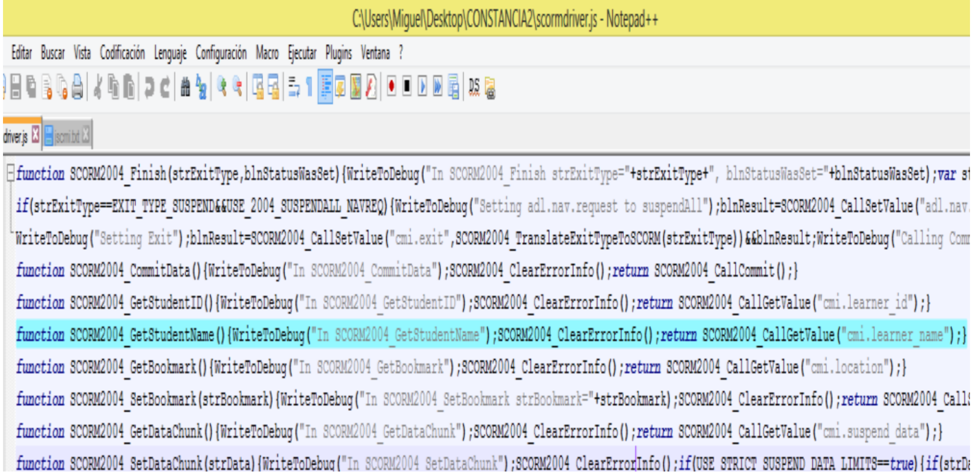
In figure 2, a screenshot of the configuration process of advanced sequencing rules (SCORM 2004) with ReLoad Editor software (from ADL), is presented.

When loading the M-SCO sequence, the Moodle LMS established rules are skipped and the user will have access to all LO’s of the M-SCO, this presents a big problem if an LO is an assessment, because students may review any questions and then check other LOs to get the answer. This is due to the LMS platform supporting only SCORM version 1.2 and not having the necessary cmi elements to restrict the LOs of the M-SCO. How to sequence an M-SCO by means of the Reload Editor is detailed in [5].

4 Capture of tracking elements in the LMS´s LO

In order to obtain an advanced sequencing of the LOs in an M-SCO with SCORM version 1.2, it is proposed to use the cmi elements of each LO, with it a set of rules within an M-SCO which would be configured portable (being inside the .zip file or LO) and independent of each LMS platform, plus no modification of LMS would be required (with the installation of a system), since the M-SCO contains the necessary sequencing rules.

As mentioned above, the cmi elements are located in the scormdriver.js document, however to access them, it is necessary to call the function that contains them. For example, in scormdriver.js the "GetStudentName ()" function contains the element "cmi.learner_name". Figure 3 shows this function.



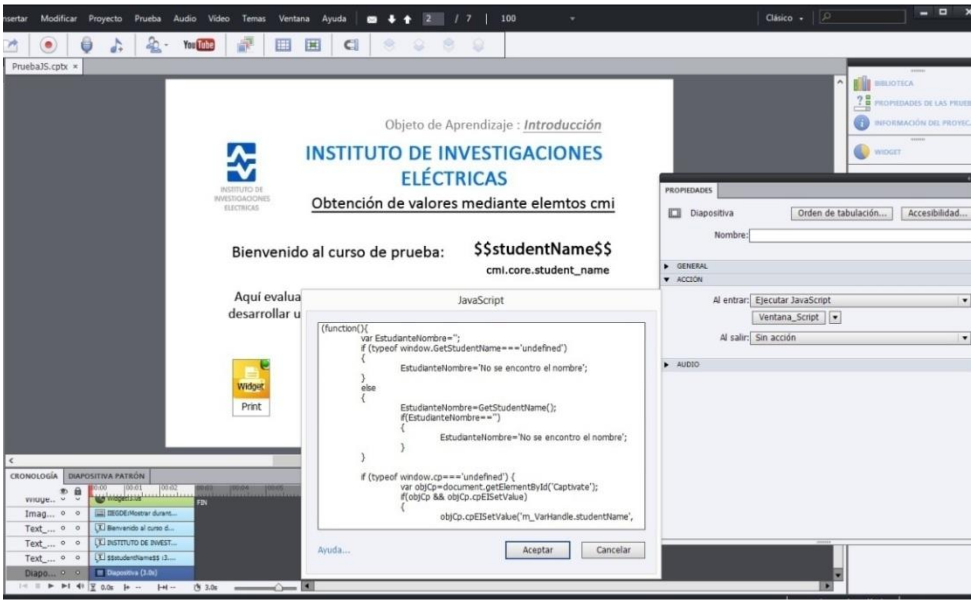
```
function SCORM2004_Finish(strExitType,blnStatusWasSet){WriteToDebug("In SCORM2004_Finish strExitType="+strExitType+", blnStatusWasSet="+blnStatusWasSet);var st
if(strExitType==EXIT_TYPE_SUSPEND&&USE_2004_SUSPENDALL_NAVREQ){WriteToDebug("Setting adl.nav.request to suspendAll");blnResult=SCORM2004_CallSetValue("adl.nav.
WriteToDebug("Setting Exit");blnResult=SCORM2004_CallSetValue("cmi.exit",SCORM2004_TranslateExitTypeToSCORM(strExitType))&&blnResult;WriteToDebug("Calling Comm
function SCORM2004_CommitData(){WriteToDebug("In SCORM2004_CommitData");SCORM2004_ClearErrorInfo();return SCORM2004_CallCommit();}
function SCORM2004_GetStudentID(){WriteToDebug("In SCORM2004_GetStudentID");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.learner_id");}
function SCORM2004_GetStudentName(){WriteToDebug("In SCORM2004_GetStudentName");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.learner_name");}
function SCORM2004_GetBookmark(){WriteToDebug("In SCORM2004_GetBookmark");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.location");}
function SCORM2004_SetBookmark(strBookmark){WriteToDebug("In SCORM2004_SetBookmark strBookmark="+strBookmark);SCORM2004_ClearErrorInfo();return SCORM2004_CallS
function SCORM2004_GetDataChunk(){WriteToDebug("In SCORM2004_GetDataChunk");SCORM2004_ClearErrorInfo();return SCORM2004_CallGetValue("cmi.suspend_data");}
function SCORM2004_SetDataChunk(strData){WriteToDebug("In SCORM2004_SetDataChunk");SCORM2004_ClearErrorInfo();if(USE_STRICT_SUSPEND_DATA_LIMITS==true){if(strDa
```

Fig.3. Function with which the student's name is obtained.

With the function mentioned, the student's name is obtained from the LMS, this variable could now be use (if necessary) in some LO, for example if required to grant a certificate to a user successfully passing a course. The "cmi.core.score.raw ()" function gets the user qualification obtained in any assessment.

5 How to Get the cmi Values Using Javascript

Authoring tools (Captivate in our case) have the option of adding a block of JavaScript code in a particular "slide". The inclusion of this type of code with the authoring tool Captivate can be appreciated in Figure 4.



Objeto de Aprendizaje : *Introducción*

INSTITUTO DE INVESTIGACIONES ELÉCTRICAS

Obtención de valores mediante elemtos cmi

Bienvenido al curso de prueba: $$studentName$$

cmi.core.student_name

Aquí evalua
desarrollar u

JavaScript

```
(function(){
        var EstudianteNombre='';
        if (typeof window.GetStudentName==='undefined')
        {
                EstudianteNombre='No se encontro el nombre';
        }
        else
        {
                EstudianteNombre=GetStudentName();
                if(EstudianteNombre=='')
                {
                        EstudianteNombre='No se encontro el nombre';
                }
        }

        if (typeof window.cp==='undefined') {
                var objCp=document.getElementById('Captivate');
                if(objCp && objCp.cpEISetValue)
                {
                        objCp.cpEISetValue('m_VarHandle.studentName',
```

PROPIEDADES

Diapositiva | Orden de tabulación... | Accesibilidad...

Nombre:

GENERAL

ACCIÓN

Al entrar: Ejecutar JavaScript

Ventana_Script

Al salir: Sin acción

AUDIO

Fig.4. Adding JavaScript in Captivate.

The function providing the cmi values is shown in the following code block:

```
(function(){
        varEstudianteNombre='';
        if (typeofwindow.GetStudentName==='undefined')      {
                EstudianteNombre='No se encontro el nombre';
        }
        else        {
                EstudianteNombre=GetStudentName();
                if(EstudianteNombre==''){
                        EstudianteNombre='No se encontro el nombre';}
        }

        if (typeofwindow.cp==='undefined') {
                varobjCp=document.getElementById('Captivate');
                if(objCp&&objCp.cpEISetValue)   {
                        objCp.cpEISetValue('m_VarHandle.studentName',  EstudianteNombre);}
        }
        else {
                if(cp.vm && cp.vm.setVariableValue){
                        cp.vm.setVariableValue('studentName', EstudianteNombre);        }
        }
})();
```

The line “*EstudianteNombre=GetStudentName();*” is divided in two parts:

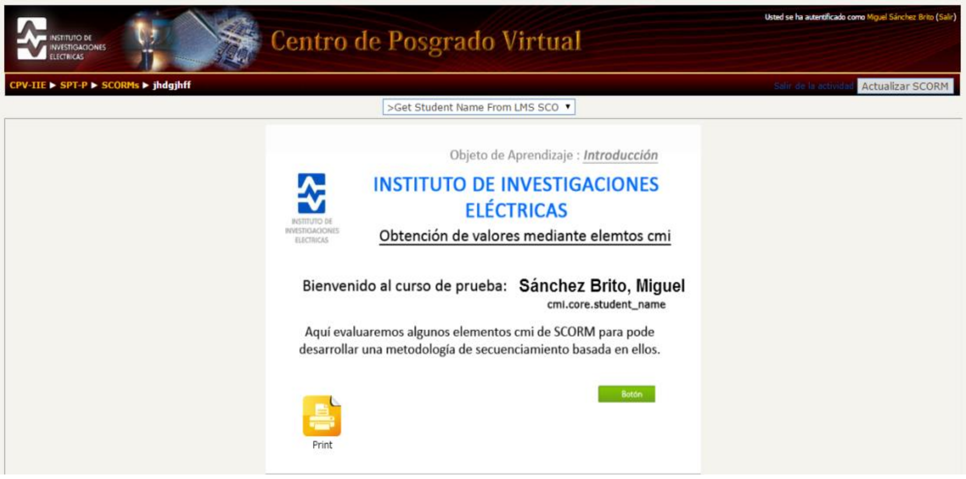
– ***GetStudentName***(): It is the most important line within the code as it allows us to access the functions contained in the cmi elements, this function is changing as we can get different cmi values.

– ***EstudianteNombre****:* It is a variable which is assigned the value obtained by the above function.

A complete list of cmi elements used in the SCORM standard is to be found in [2].

6 Results: Performance Evaluation of the Code

To verify the correct operation of the above code, by way of example an LO was developed in Captivate and mounted in the Moodle LMS of the Virtual Graduate Center of the Institute of Electrical Research (CPV-IIE).


INSTITUTO DE INVESTIGACIONES ELECTRICAS
Centro de Posgrado Virtual
Usted se ha autentificado como Miguel Sánchez Brito (Salir)
CPV-IIE ▶ SPT-P ▶ SCORMs ▶ jhdgjhff
Salir de la actividad
Actualizar SCORM
>Get Student Name From LMS SCO ▾
Botón
Print

Objeto de Aprendizaje : *Introducción*

INSTITUTO DE INVESTIGACIONES ELÉCTRICAS

Obtención de valores mediante elemtos cmi

Bienvenido al curso de prueba: **Sánchez Brito, Miguel**
cmi.core.student_name

Aquí evaluaremos algunos elementos cmi de SCORM para pode desarrollar una metodología de secuenciamiento basada en ellos.

Fig.5. Getting the name from the LMS platform.

Using "GetStudentName ()" we get the name of the user of the LMS platform, in Figure 5 we can see that the function performs its task properly.

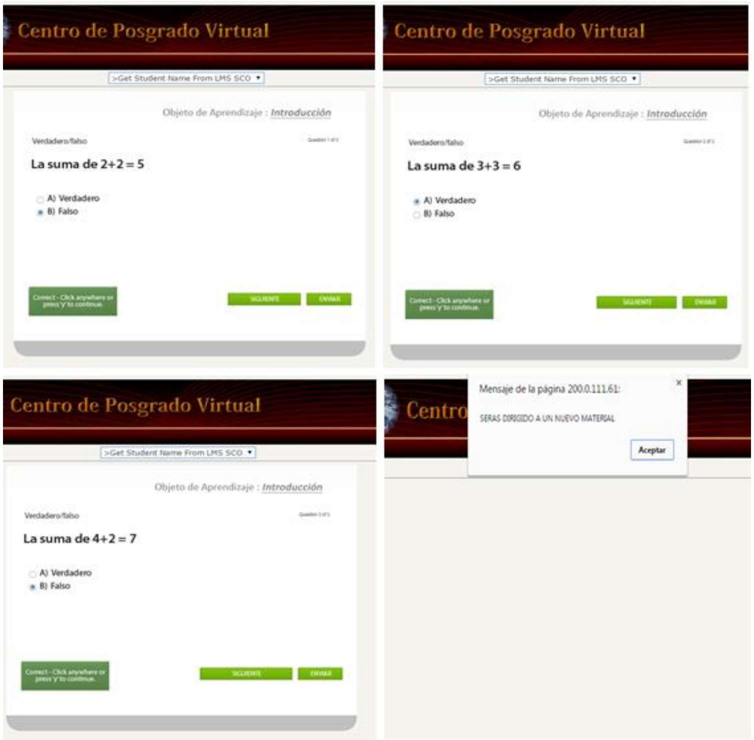
One of the most important aspects for decision-making in any learning system is the score on any assessment. Because of that a test assessment which consists of only three questions was developed. Using *"cmi.core.score.raw ()"* we access the grade obtained by the user, and with the following line of code we developed a restriction for access to the next LO within the M-SCO:

```
if(EstudianteCalif>60)
        {
                    window.open("Alerta.html","_self");
        }
```

The logic in the test is as follows:

If the user has two or more correct answers (it is equivalent to having a share of about 66% satisfaction, as there are 3 questions equivalent to 100%), the rule set for this example will launch a message warning the user he/she will be directed to new material.

Figure 6 shows the rule determined works correctly.



Centro de Posgrado Virtual
>Get Student Name From LMS SCO
Objeto de Aprendizaje : Introducción
Verdadero/falso
Correct - Click anywhere or press 'y' to continue.
SIGUIENTE
ENVIAR
Aceptar

La suma de 2+2 = 5

○ A) Verdadero
● B) Falso

La suma de 3+3 = 6

● A) Verdadero
○ B) Falso

La suma de 4+2 = 7

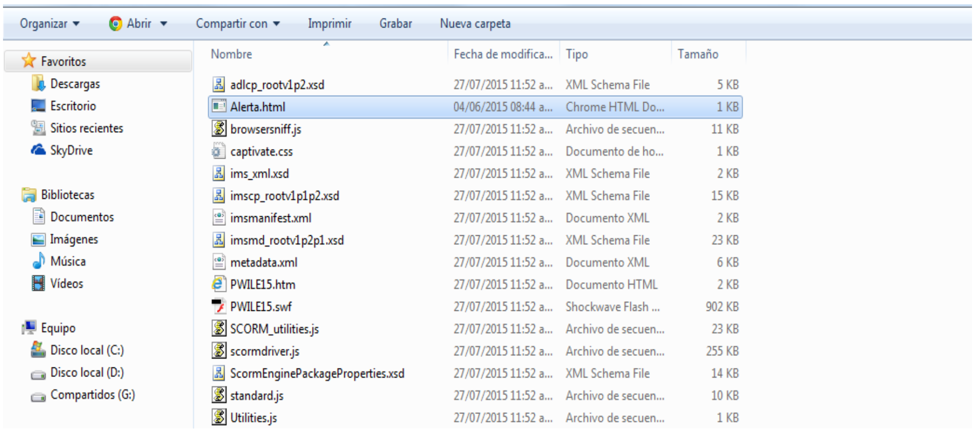
○ A) Verdadero
● B) Falso

Mensaje de la página 200.0.111.61:

SERAS DIRIGIDO A UN NUEVO MATERIAL

Fig.6. Assessment score detection by means of cmi elements.

Figure 7 shows the location of the file "Alerta.html" within the M-SCO, which contains the code above.



Nombre	Fecha de modifica...	Tipo	Tamaño
adlcp_rootv1p2.xsd	27/07/2015 11:52 a...	XML Schema File	5 KB
Alerta.html	04/06/2015 08:44 a...	Chrome HTML Do...	1 KB
browsersniff.js	27/07/2015 11:52 a...	Archivo de secuen...	11 KB
captivate.css	27/07/2015 11:52 a...	Documento de ho...	1 KB
ims_xml.xsd	27/07/2015 11:52 a...	XML Schema File	2 KB
imscp_rootv1p1p2.xsd	27/07/2015 11:52 a...	XML Schema File	15 KB
imsmanifest.xml	27/07/2015 11:52 a...	Documento XML	2 KB
imsmd_rootv1p2p1.xsd	27/07/2015 11:52 a...	XML Schema File	23 KB
metadata.xml	27/07/2015 11:52 a...	Documento XML	6 KB
PWILE15.htm	27/07/2015 11:52 a...	Documento HTML	2 KB
PWILE15.swf	27/07/2015 11:52 a...	Shockwave Flash ...	902 KB
SCORM_utilities.js	27/07/2015 11:52 a...	Archivo de secuen...	23 KB
scormdriver.js	27/07/2015 11:52 a...	Archivo de secuen...	255 KB
ScormEnginePackageProperties.xsd	27/07/2015 11:52 a...	XML Schema File	14 KB
standard.js	27/07/2015 11:52 a...	Archivo de secuen...	10 KB
Utilities.js	27/07/2015 11:52 a...	Archivo de secuen...	1 KB

Fig.7. Location of JavaScript code to execute actions based on cmi elements of the SCORM standard.

With the latter, the use of the cmi elements for sequencing [6, 7] of the training material according to the performance of each student is demonstrated, thus being able to develop an advanced sequencing of (reusable) learning objects that meet the SCORM standard.

7 Conclusions

In the present research work a way to leverage the cmi elements of a learning object was exhibited in order to use it in any tutoring system. The tests presented here are evidence of the proper functioning of the code developed.

It is now possible to sequence learning objects in an advanced way and display them in an LMS, based on the SCORM standard for e-learning. This brings us one step closer to automating online courses delivered through an LMS, where the advanced sequencing is the last step of the intelligent tutoring.

8 Future Work

As future work we propose to develop an M-SCO containing several LOs and develop an intelligent system based on cmi and JavaScript elements, the latter would be very useful since it is practically a portable intelligent system when in the M-SCO with the capability of complying with the SCORM standard.

References

1. Miguel Sánchez-Brito, José Ruiz-Ascencio, Carlos Felipe García-Hernández: SCORM Cloud for an advanced sequencing of learning objects on LMS Moodle platform. Research in Computer Science, 87, pp. 19–26 (2014)
2. Advanced Distributed Learning, http://www.adlnet.org (2015)
3. Rustici Software, http://www.scorm.com (2015)
4. Moodle, http://www.moodle.org (2015)
5. C. F. García-Hernández, M. Sánchez-Brito, F. F. Jiménez-Fraustro: Configuración de la Plantilla 10 y Modelo Educativo 3 de ADL: Secuenciamiento con Remediación. In: X Congreso Internacional sobre Innovación y Desarrollo Tecnológico CIINDET, No. 620, Cuernavaca, Morelos, México, pp. 1–7 (2013)
6. Ahmad A. Anbar, Ahmad M. Al-Shishtawy, Marwa Al-Shandawely, Tamer A. Mostafa, Adham Bolbol, Ahmed Hammad, Saad Sunoallah, James Everett, Kutluk Özgüven: Applying Pedagogical Concepts in Online Course Development: Experiences from the Mediterranean Virtual University, pp. 1–11 (2005)
7. Guillermo Rodríguez Ortiz, Ricardo Molina González, Carlos Felipe García Hernández, Mayra Jazmín Mendoza Bahena: Metodología para la implantación de la capacitación a distancia. In: XII Congreso Internacional sobre Innovación y Desarrollo Tecnológico CIINDET, No. 7, Cuernavaca, Morelos, México, pp. 1–9 (2015)