Adaptive Learning Sequencing for Course Customization: A Web Service Approach

Ivan Madjarov and Abdelkader Betari

Abstract — We discuss a learning model that enables the creation of optimal learning strategies that suit learners’ needs. A customized learning content is delivered to learners as managed sequences of learning activities which form a hierarchical structure that can be modeled by means of Petri Nets. Learning content customization is a dynamic process supported by a set of standards-based data models. The approach we adopted is implemented as Web services in a service-based learning architecture. Learning objects are sequenced according to learners’ responses to exercises and questions.

Index Terms — Adaptive sequencing, Learning activity, Petri Nets, Web services.

I. INTRODUCTION

A customized learning activity is a process of dynamic assembly of content objects, extracted from content repositories to enable the creation of an optimal learning strategy that suits learners’ needs. As defined in [4] there are four patterns of processing a customized learning activity: (1) a repeat random selection of Learning Objects (LOs) until the learner demonstrates ability; (2) choosing of next Learning Object (LO) is based on learner’s actions or responses to questions; (3) based on a learner’s profile a set of LOs are chosen; (4) finding a LO that best meets a given learning objective and discover current learner’s skill set.

A collection of specifications is adapted from multiple sources to provide a comprehensive suite of e-Learning capabilities that enable interoperability, accessibility and reusability of Web-based learning content. Thus, SCORM [1] supports a detailed description of Assets, SCO (Sharable Content Objects) and LOs using LOM [11] metadata specification with IMS metadata elements [12]. SCORM describes the content components (Content Aggregation Model) used in a learning activity, the way of packaging those components (Content Package) for information exchange between systems, and metadata descriptions (IMS manifest) of those components to enable search and discovery. The SCORM Run-Time Environment provides also a means for interoperability between LOs and LMS (Learning Management System).

SCORM SN (Sequencing and Navigation) is based on the IMS-SS (Simple Sequencing) Specification [14] which defines a method for representing the intended behavior of an authored learning activity such that any conformant LMS will be able to sequence discrete content components in an AT (Activity Tree), based on the results of a learner’s interactions [28]. This enables to control the conditions under which a LO is selected, delivered, or skipped during presentation to a learner. Typically, this functionality is used to test a learner’s understanding and re-route them to remedial content if a test pass mark is not achieved. However, it must be acknowledged that structures with complicated sequencing rules in the AT make the design of course sequences rather difficult [17], [18]. By following this approach, several issues can be observed [7] e. g. high development costs due to a limited reusability across different standards, learning contexts and learning platforms. To overcome these issues, our approach in this paper changes this data-centric paradigm to a dynamic metadata-based service-oriented approach.

To help authors understand the tricky sequencing behavior rules in SN, a course can be represented as a graph. We propose a systematic approach to construct a course that can be represented as Petri Nets (PNs) model to simulate the learning behavior, partially developed in [28]. PNs are a powerful formal tool for system modeling and validation and can be applied to model basic sequencing rules for designing a course with complex sequencing behaviors. The model described in this paper is complemented with a Web Services-based architecture for learning customization by response.

The remainder of this paper is as follows. Section 2 presents the state of the art in content aggregation and content sequencing. Section 3 presents our PNs model approach for adaptive LOs sequencing, while Section 4 presents an implementation. Finally, concluding remarks are given in Section 5.

II. BASIC CONCEPTS

The functional building block of SCORM is an asset which is defined as the basic form of learning digital content. An asset is any electronic representation of media such as text, images, sound, video, flash animation object, questionnaire,
Web page, or any other piece of data. A collection of one or more assets is named SCO (Shareable Content Objects) and makes up a LO [11].

A. The Content Aggregation Model

A collection of SCOs represents the level of granularity for learning content pieces about which a LMS (Learning Management System) or a LCMS (Learning Content Management System) tracks detailed information. A SCO is the learning unit able to communicate with the e-Learning platform in using APIs or functions (Javascript or ECMA Script) defined by SCORM. An LMS must provide an implementation of the SCORM API for use by LOs.

LOs [10] are building blocks of study, exercise, or practice for a solution that ensures interoperability and reusability problems since blocks relies on existing standards like SCORM [1], IMS [11], LOM [10]. LOs can be authored independently of the delivery media by using an authoring system, they may be stored inside or outside a LCMS, and they could be distributed on demand over the Web [25]. This strategy is in agreement with the first goal of SCORM to separate learning content from learning platforms. This allows learning content to be developed once and run on any e-Learning platform in compliance with standards.

The process of content aggregation allows SCOs to be bundled together to create a learning experience. This bundle includes an XML file that describes the contents of the package and the order in which the SCOs are to be delivered and find out. These descriptions belong to IMS-CP (Content Packaging) Specification [11] and the IEEE LOM (Learning Object Metadata) Standard [10]. An IMS-CP is a Zip file that contains a collection of learning resources, an XML manifest file with a list of available resources, pointers to them, and one or more sets of instructions for structuring and/or sequencing available LOs into a coherent learning experience.

B. Sequencing and Navigation Model

The SCORM SN model allows a course designer to define Learning Activities (LAs), which may be conditionally sequenced based on information derived by tracking the learner's progress. The key concept of the SN model is the Activity Tree (AT). Items in a CP are mapped by the LMS to LAs in the AT with SCOs. The course author annotates the CP with sequencing rules, so the LMS can use the AT to sequence a course at run-time.

Each sequencing rule has an "if-then" structure, where the "if" conditions use the learner's tracking information. The "then" sequencing actions include eliminating a node or a node's children from being sequencing candidates, exiting a parent node or all nodes above that node, retrying the current one or all nodes, and sequencing the next or previous nodes in a pre-order traversal. These sequencing rules may be used to create a continuum of controlled sequencing. At one extreme of the continuum, the learner is free to choose any learning activity in any order. At the other end, the learner can only move forward, and tracking information is used so that the learner skips over material they already know [29].

IMS-SS Specification defines a model based on an enumeration of sequencing and behavior rules. The content is organized as a hierarchy with multiple objectives defined per LO. Thus, this specification defines a method for representing the intended behavior and a relative order in which elements of content are to be presented to the learner and the conditions under which a piece of content is selected, delivered, or skipped during this presentation. IMS-SS incorporates rules that describe the branching or flow of LAs through content according to the outcomes of a learner's interactions with content and can be used to sequence all types of content. Sequencing rules in IMS-SS are used to impact the order of activities presented to the learner. Limit conditions, such as attempt limits, duration limits and date limits, are used by the sequencing rules to further decide which activity is sequenced next to a learner. Sequencing rules and limit conditions are part of the IMS-SS definition model that defines the vocabulary, semantics and values required to execute IMS-SS behaviors. The definition model has an XML binding that can be incorporated into an IMS-CP manifest file (imsmanifest.xml).

C. Learning activities managed through simple sequencing

As defined in [13] a Learning Activity (LA) is a “pedagogically neutral unit of instruction, knowledge, and assessment”. The assessment process [3] can be seen as an activity that completes the e-Learning cycle: giving the results, and support for the feedback to the students, updating the contents of the learning system. A root LA can have sub-activities and may be nested to an arbitrarily deep level. Each activity may have a tracking status associated for each learner. Activities may be launched many times; they can be suspended, abandoned, or exited normally. All activities are performed within the context of a parent activity.

Activities managed through IMS-SS are defined as an AT. Learning processes are described in terms of traversing the nodes of the tree to determine the activity delivered to the learner. The tree is not necessarily balanced and its branches are not necessarily of equal length [20]. In Fig. 1, we consider activity (c) as part of activity (a), and activity (f) as part of activity (d), which is part of activity (b), and which is part of activity (a). We can assign some pedagogical tag to those activities. Thus, we can assign (a) to a Course, (c) to a Module A, (d) to an Exercise A, and (g) to a Question A3. Other assignments are always possible.

The IMS-SS specification defines in [13] the canonical traversal of the AT as pre-order traversal where some activities are associated with other activities into a hierarchy. A parent activity and its children are referred to as a cluster of activities. Clusters may have sequencing rules and limit conditions associated with them. The example in Fig. 1 shows that starting from Course, the traversal would go to Intro, then Module A, then Exercise A, then Question A1, then Question A2, then Question A3, then Module B, then Exercise B, etc. Note that the reverse pre-order traversal is not just a mirror image of the forward pre-order traversal. In Fig. 1 the precedent of Module B is Question A3, not Module A. In some
situations this canonical traversal can cause various problems of logical sequencing as it is the case in our example. The IMS SS model consists of static rules applied to the content tree.

![Fig. 1. An example of an IMS sequencing activity tree](image)

An Activity Resource may be associated with any node in the AT that is a child-node. This means that the delivery of an activity involves the delivery of a LO. Some LOs may be able to communicate results, and others not. IMS-SS [14] does not specify how this data is communicated. In the example above, we use LOs of different natures that communicate results to the next node.

![Fig. 2. Content packaging model](image)

In this paper we are interested in the study of a more precise use-case scenario: upon initial connection, the learner receives the content of a first LA consisting in an Intro page (see Fig. 1). After receiving this page, the learner goes on to the next LA – Module A that is a page with a learning content. When exiting this page another activity of type Exercise is sent to the learner. Depending on learner’s answer to this activity a suite of LAs of type Question can be addressed to him. This pedagogical strategy aims to evaluate the level of his or her understanding of selected pieces of learning content. We denote these selected pieces of learning content as a Learning Cluster (LC). A LC may contain different types of learning content (Text page, Exercise, Question), which may return different kinds of results (boolean or numeric). In our strategy the next proposed LC depends on learner’s score of the previous LC and so on.

A cluster of LCs is a possible solution for the mentioned problems. A cluster is an organized aggregation of LAs consisting of a single parent activity and its first level children. A cluster may include various activities presented as Cluster Activity Tree (CAT). The parent activity of a cluster contains the information about the sequencing strategy for the cluster that may be different for another cluster present in the course AT. With this structure we propose that a CAT may remain independent with a set of elements that can be used to describe and affect various sequencing behaviors. Thus, the defined sequencing rules are applied to only one CAT. Automatically this suggests the possibility of having various sequencing rules for different CAT. The sequencing rules for LCs sequencing are defined in another high-level of organization in IMS CP (Fig. 2). In the contrary case the description of sequencing rules on one level of organization becomes very complex to build and its implementation very complicated to interpret in a dedicated LMS. In some related works, which tackle this problem, the proposed solutions are based on the use of hierarchical or sequencing graphs [28] and swarm intelligence techniques [7], or an object-oriented model for course composition [17]. The graphs define transitions between different learning activities based on some parameters defined by the course author according to learner’s actions. There are some limitations when it comes to maintaining and updating learning activities with new content [28]. Finally sequencing graphs are exported to IMS-SS semantics on one level of organization. The sequencing component analysis in [17] is similar to our work with the difference that we organize the LOs into CAT in synchronized mode.

Our novel approach for manifest construction is illustrated in Fig. 2. What we are trying is to reduce the complexity level of sequencing rules description and implementation. In doing so we can extend the IMS-SS specification capacities where the sequencing information can be associated with any organization and/or with any item within. A resource is associated with any LO. The mapping to the LOs is unique and should not be changed from the logical point of view. Organizations can be defined in a hierarchical way using items (CAs) and sub-items (CATs). Thus, we can obtain a set of related organizations which will be indicated in several sub-manifests who make the insmanifest file in the CP. A consequence with that is the possibility of cluster exchanging in real time according to adaptive criteria. Sequencing rules can be defined between clusters in compliance with the IMS-SS Specification.

III. MAPPING PETRI NETS ONTO LEARNING OBJECT SEQUENCES

As mentioned above learning activities are defined in IMS-SS specification as an AT i.e., a graph. A Petri Net (PN) is a directed, connected and bipartite graph in which each node is either a place or a transition. The arcs of this graph connect the transitions to the places or the places to the transitions. The places contain tokens or marks, which go from place to place by crossing the transitions according to crossing rules.

We are interested in devising an IMS-SS model liable for the sequencing of course-related Web pages to learners according to their tests results by defining analysis rules. Thus, the management of page sequencing is based on an analysis of incoming events via a PN model. PNs provide a well-known process modeling technique with formal semantics, which can be used to model and analyze several types of systems (e.g., sequencing behaviors [17] and Web services [8]).
A. Course Sequencing Behavior Model

Formally, a PN [19] is defined as a flow relation. The triple \( N \in (P,T,W) \) is a PN where: \( P \) is a finite set of places, \( T \) is a finite set of transitions and \( W \) is a set of directed arcs (flow relation). The flow relation may also be quantified in terms of an evaluation function \( W : (P \times T) \cup (T \times P) \to \{0,1\} \). It is common practice to draw ordinary places by circles and transitions by squares.

For a large number of entities that present similar behaviors, the use of a colored net makes possible to condense the model. The colored PN are nets in which the tokens carry colors. It is in particular well suited for systems that consist of a number of processes, which communicate and synchronize.

We adopt a traditional Petri Net as defined in [6], with refinement and changes. Our PN model is defined as synchronous and colored PN and it is represented as \( R = (P,T,Col,\pi,\lambda) \); where \( P \) is a finite and non empty set of places that consist of two subset: \( p_1, p_2, \ldots, p_m \) (ordinary place subset) and \( c_1, c_2, \ldots, c_n \) (control place subset); \( T \) is a finite and non empty set of transitions; \( Col \) is a non empty set of color properties; \( \pi \subseteq P \times T \) is the input place, \( \lambda \subseteq T \times P \) is the output place, and we have the property: \( P \cap T = \emptyset \). Control places we draw by double circles.

According to [17] we can introduce some typical interpretations of \( P \) such as preconditions, input data, conditions, and resource needed. The ordinary places represent learning material including lessons, exercises, tests or questionnaires (LOs) and control places give assistance to model management (analysis rules). The transitions (\( T \)) represent events, computation steps, or state changing operators. Arc directs the information flow outgoing form input place as either a point of departure or a temporary pause. The dynamic sequence behavior of model is simulated by the firing rules. A transition will be fired if the token of input place will be moved to the output place according to the weight of output arc.

To address the problem and the solution set forth in this article, we propose to study the sequencing control of a Web-based course. Based on the scenario already discussed in section 2.2 we present the following example. Let us assume that a course is composed of three main pages, three exercises and nine questions (i.e. LOs). The course structure is divided in three clusters. A cluster is composed of three different kinds of LAs: one main page, one exercise and, optionally, a suite of three questions, or another kind of LOs. The next proposed learning cluster depends on learner’s score of the previous one and so on. The score is communicated to the control places which determine the next sequencing behavior. Inside a cluster, one main exercise is associated to each course page and a sequence of three questions is associated to each exercise. Upon initial connection, the learner receives the content of the first main page (Page 1). After receiving this page, the learner goes on to the first exercise (Ex. 1), which corresponds to an event of type “next”. The learner’s answer to the exercise represents an event of type “validate”. The answer is analyzed and results in a constant from the set \{True, False\}. If the answer is correct, the next (cluster) course page is sent to the learner (Page 2). If the answer is not correct, a question is sent to the learner (Q 1-1). If the learner’s response to the question is correct, the second main page of the course is sent to the learner, if not the learner receives a second question of the same type as the previous one (Q 2-1). If the answer of this second question is correct, the next course page is sent to the learner, otherwise the learner receives a third question of the same type as the two previous ones (Q 3-1). The same process applies to all course pages (clusters) and their corresponding learning sequences (LOs). This pedagogical strategy contributes to evaluate the level of learner’s understanding.

![Fig. 3. The course cluster sequencing behavior presented by a Petri Net](image)

The PN model associated with this example is represented on Fig. 3. Crossing orders are indicated on the corresponding transitions. If the learner fails all the exercises and questions corresponding to a course page, a start page is sent back. Similarly, once a learner successfully completes the third main exercise, a final page is sent back. For simplification purpose, these ending conditions are not represented.

In our approach a course may be represented as a 5-tuple: \( C = (NC, DS, URI, CS, PN) \) where NC is the name of the course, DS is the description of the service provided, URI is the address of the course, CS is a set of component service-functions attached to a Web service [8], PN is the PN which models the course sequence.

The relationship between traditional PN formalism and course elements is as follows: the \( P \) set represent the set of course LOs and control analysis objects, the \( T \) set represent the set of events corresponding to the answers, and the color represents learner’s identity. We build the PN by associating a place for each LO and a transition to any LO changes. To manage the LO changes, we analyze the data coming from the learner in control places. For this purpose, we introduce two complementary functions:

- \( RepEx(a) \) is a function that defines the type of data coming from the learner activity (“validation” or “next”),

...
• \( \mu(a) \) is a function that analyzes a response of type "validation". This service translates the learner answer to a constant \{True, False, Next\}.

The analysis function compares the returned answer with a standard answer. By evaluating their differences, a grade \( \text{note}(x) \) is generated as an integer value between 0 and 10 (0 \( \leq \text{note}(x) \leq 10 \)):

\[
\begin{align*}
\mu(x) &= \text{True} \{ (\text{RepEx}(x) = \text{True}) \wedge (\text{note}(x) \geq 5) \}, \\
\mu(x) &= \text{False} \{ (\text{RepEx}(x) = \text{True}) \wedge (\text{note}(x) < 5) \}, \\
\mu(x) &= \text{Next} \{ \text{RepEx}(x) = \text{False} \}.
\end{align*}
\]

Any transition in the PN under consideration is ordered according to one single place. This justifies its encoding using a Hash table with collisions control.

B. Resource Mapping

The course configuration is described by the author in the resource section of the imsmanifest XML document. The content of this resource description conforms to the following schema:

```xml
<xml version="1.0"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
  <xsd:element name="valid">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PlaceE" type="xsd:string" />
        <xsd:element name="Transition" type="xsd:string" />
        <xsd:element name="PlaceS" type="xsd:string" />
        <xsd:element name="URL" type="xsd:string" />
        <xsd:element name="UDDI" type="xsd:string" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

where \( \text{PlaceE} \) and \( \text{PlaceS} \in \text{LOs}, \text{Transition} \in \{ \text{True, False, Suite} \} \).

The key in the table is communicated by concatenating "\( \text{PlaceE} + \text{Transition} \)"; the corresponding value is obtained by concatenating "\( \text{PlaceS} + \& + \text{URL} \)."

"\( \text{PlaceE} \)" is the entry point of "\( \text{Transition} \)." For the validation of "\( \text{Transition} \)," a mark must be present in "\( \text{PlaceS} \)," and the value of the "\( \text{Transition} \)" must correspond to the value of \( \mu(x) \). In other words, when a learner interacts with the LO corresponding to "\( \text{PlaceE} \)" and "\( \text{Transition} \)" is validated by the event corresponding to the answer, the following LO becomes "\( \text{PlaceS} \).

With this interaction scheme we can react in an adapted way according to the learner’s answer quality. Thus, according to the learner’s score, obtained from tests in a given stage, the associated service can request a LO that is more adapted to its level of knowledge. Our objective with this approach is first, to favor more dynamism in the static descriptions of sequencing behaviors, and second to ensure easy integration of various types of LOs in the schema of sequencing rules defined by IMS-SS (e.g. IMS-QTI [16] and IMS-SS).

IV. IMPLEMENTATION

A. IMS Web Services Background

IMS General Web Services [12] (IMS-GWS) specification promotes interoperability across Web Service-based specification implementations on different e-Learning platforms. This specification identifies a core set of specifications that are to be used to implement a Service-Oriented Architecture (SOA). The IMS-GWS specification addresses interoperability in the application layer, in particular, the description of behaviors exposed via Web Services. IMS-GWS specifications are produced to ensure that all of the services defined in all of the IMS specifications use a common, and thus compatible, message exchange infrastructure. A consequence of this is that the creation of a service specification is focused on the interaction process and the service methods required which realize that process.

A traditional IMS-SS implementation could create a single monolithic sequencing engine or a complete sequencing service. This monolithic service, in fact, is composed of a collection of low-level sequencing services. Each of these sub-services implement a part of the sequencing behavior defined in the IMS-SS specification. Based on SOA concept each sub-service can be implemented as a stand-alone application in conjunction with a set of external applications managed by a WSMS (Web Services Management System) [25]. Web services that comprise a distributed navigation and sequencing system may include: navigation service, sequencing service, delivery service, rollup service, remediation service, and exit service [21]. For course authors, course administrators and learners needs these Web services can be individually assembled by using distributed components to provide the functionality defined by a sequencing rule in a CAT, or in a cluster tree.

B. Our Implementation Scenario

We are interested in devising a Web service [2], [5] responsible for the sequencing of course-related Web pages to learners according to their tests’ results. Two Web service functions are used to define the type of the data coming from the learner’s machine and to analyze his response. The learner’s answer is translated to the analysis rule service to determine the type and the propriety of the next proposed cluster of LOs (Fig. 3). With this we can react in an adapted way according to the learner’s answer quality. The associated Web service can subject a LO that is more adapted to its level of knowledge. With this approach we encourage more dynamism in the static descriptions of sequencing behaviors and second we ensure more easily, without constraints, the integration of various types of LOs in the run-time schema of sequencing rules.

Our "Xesop-Jaxe" system is still under development [23], [24], [25]. It is based on SOA principles so that the pedagogical content is encapsulated inside a Web service in order to increase interoperability and reusability not only for learning data but applications too. The system consists, among others, of an Open Semantic Editor Suite (OSES) that is a set
of XML-based tools (editor and plug-ins) for creating, editing, and storing learning content in a native XML database (NXDB). XML is used for encoding textual and non-textual information such as vector graphics, mathematical expressions, synchronized multimedia documents, complex forms, quizzes, etc [23], [24]. The role of SOA we suggest is to integrate existing free and open source LCMS, in which different external components are implemented by a WSMS. For course authors, course administrators and learners, e-Learning systems can be individually assembled by using distributed components to provide the functionality they really need. In addition, we propose a new Web service-based client in an e-Learning system in using AJAX technique [25] for a better client interactions management with exercises or quizzes.

V. CONCLUSION

Learning content customization is composed of learning activities that are delivered to learner as managed sequences. The PNs-based model that we propose consists of a hierarchical structure of learning activities. LOs are organized in a cluster tree and are sequenced according to learner’s actions or responses of exercises and a suite of questions to discover current learner’s skill set. With this interaction schema we can react in an adapted way according to the learner’s score, obtained from tests in a given stage. The associated service can subject a LO that is more adapted to its level of knowledge. This approach allows more dynamism in the static descriptions of sequencing behaviors and ensures easy integration of various types of LOs in the sequencing rules schema. The approach is being implemented as a Web service-based learning infrastructure.

REFERENCES