Adaptive Instructional Planning using Ontologies

Pythagoras Karampiperis, Demetrios Sampson
Advanced e-Services for the Knowledge Society Research Unit,
Informatics and Telematics Institute,
Centre for Research and Technology Hellas,
42, Arkadias Street, Athens, GR-15234 Greece
e-mail:{pythk, sampson}@iti.gr

Abstract

Adaptive instructional planning or sequencing is recognized as one of the most interesting research questions in intelligent learning management systems. In this paper, we address the adaptive learning object sequencing problem in intelligent learning management systems proposing a concrete methodology based on the use of ontologies and learning object metadata. The result is a generic Instructional planner capable of serving for both Adaptive and Dynamic Courseware Generation.

1. Introduction

Intelligent learning management systems seek to provide adaptive navigation and adaptive sequencing. Adaptive navigation seeks to present the content associated with an on-line course in an optimized order, where the optimization criteria takes into consideration the learner’s background and performance on related knowledge domain [1], whereas adaptive sequencing is defined as the process that selects learning objects from a digital repository and sequence it in a way which is appropriate for the targeted learning community or individuals [2]. Adaptive sequencing of learning objects is recognized as one of the most interesting research questions in intelligent learning management systems [3, 4].

Although many types of intelligent learning systems are available, we can identify five key components which are common in most systems, namely, the student model, the expert model, the pedagogical module, the domain knowledge module, and the communication module.

In most intelligent learning systems that incorporate course sequencing techniques, the pedagogical module is responsible for setting the principles of instructional planning based on a set of teaching rules according to the learning preferences of the learners [4]. In spite of the fact that most of these rules are generic (i.e. domain independent), there are no well-defined and commonly accepted rules on how the learning objects should be sequenced to make “instructional sense” [2, 5].

In this paper, we address the adaptive sequencing of learning objects in intelligent learning management systems. In the next section we discuss the main architectural approaches in automatic course sequencing. The third section discusses the main steps in the instructional planning process and proposes the use of ontologies and learning object metadata.

The forth section presents the decision framework for extracting the appropriate learning path according to the learner’s navigation steps.

Finally, we present simulation results of the proposed methodology.

2. Automatic Course Sequencing

In literature, two main approaches in automatic course sequencing have been identified [4]:

- **Adaptive Courseware Generation**, where the main idea is to generate a course suited to the needs of the learners. Instead of generating a course incrementally, as in a traditional sequencing context, the entire course is adaptively generated before presenting it to the learner.

- **Dynamic Courseware Generation**, where as in the previous approach, the goal of dynamic courseware generation is to generate an individualized course taking into account specific learning goals, as well as, the initial level of the student’s knowledge. The
difference here is that the system with dynamic generation observes and adapts to student progress during his interaction with the generated course. If the student’s performance does not meet the expectations, the course is dynamically re-planned. The benefit of this approach is that it applies as much adaptivity to an individual student as possible. Through dynamic regeneration each student is able to get a highly personalized course for his/her needs.

Both the above mentioned techniques are first using filtering to generate an initial pool of personalized learning objects that match the general requirements.

![Diagram](image_url)

**Figure 1: Generalized Architecture of Automatic Course Sequencing Techniques**

This pool is generated from both distributed and local learning object repositories, for which the appropriate access controls have been granted.

The filtering process is based on general requirements such as characteristics of the language or the media of the targeted learning objects, as well as, learner characteristics such as accessibility and competency characteristics or even historical information about related learning activities, included in the Student Model module.

The result of the filtering process falls into a virtual pool of personalized learning objects that will act as an input space for the instructional planner. Figure 1 presents a generalized architecture of the above mentioned course sequencing techniques that utilize filtering and instructional planning processes.

In the next section we will present the main steps of the instructional planning process and analyze the way that ontologies and learning object metadata can be used for effective planning.

### 3. Instructional Planning

The instructional plan of an intelligent educational system can be considered as two interconnected networks or “spaces”:

- a network of concepts (knowledge space) and
- a network of educational material (hyperspace or media space).

Accordingly, the instructional planning process involves three key steps [6]:

- structuring the knowledge
- structuring the media space
- connecting the knowledge space and the media space.

#### 3.1. Knowledge Structuring

The heart of the knowledge-based approach to developing intelligent learning management systems is a structured domain model that is composed of a set of small domain knowledge elements (DKE). Each DKE represents an elementary fragment of knowledge for the given domain. DKE concepts can be named differently in different systems—concepts, knowledge items, topics, knowledge elements, but in all the cases they denote elementary fragments of domain knowledge.

Depending on the domain, the application area, and the choice of the designer, concepts can represent bigger or smaller pieces of domain knowledge. A set of domain concepts forms a domain model. More exactly, a set of independent concepts is the simplest form of domain model.
The use of ontologies can significantly simplify the task of knowledge structuring by providing a standard-based way for knowledge representation.

Ontologies are specifications of the conceptualization and corresponding vocabulary used to describe a domain [7]. They are well-suited for describing heterogeneous, distributed and semi-structured information sources that can be found on the Web. By defining shared and common domain theories, ontologies help both people and machines to communicate concisely, supporting the exchange of semantics and not only syntax. It is therefore important that any semantic for the Web is based on an explicitly specified ontology.

Ontologies typically consist of definitions of concepts relevant for the domain, their relations, and axioms about these concepts and relationships. Several representation languages and systems are defined. A recent proposal extending RDF and RDF Schema is OWL (Ontology Web Language). OWL unifies the epistemologically rich modeling primitives of frames, the formal semantics and efficient reasoning support of description logics and mapping to the standard Web metadata language proposals. OWL is a W3C Recommendation since February 2004.

For the instructional planning process we have identified four classes of concept relationships, namely:

- “Consists of”, this class relates a concept with it’s sub-concepts
- “Similar to”, this class relates two concepts with the same semantic meaning
- “Opposite of”, this class relates a concept with another concept semantically opposite from the original one
- “Related with”, this class relates concepts that have a relation different from the above mentioned

Figure 2 presents a concept hierarchy using the four concept relationships identified.

3.2. Structuring the Media Space

In most intelligent learning systems, structuring of the media space is based on the use of learning object metadata.

More precisely, in the IEEE LOM metadata model [8], the ‘Relation’ Category, defines the relationship between a specific learning object and other learning objects, if any. The kind of relation is described by the sub-element ‘Kind’ that holds 12 predefined values based on the corresponding element of the Dublin Core Element Set.

In our case we use only four of the predefined relation values, namely:

- “is part of” / “has part”
- “references” / “is referenced by”
- “is based on” / “is basis for”
- “requires” / “is required by”

3.3. Connecting knowledge with educational material

The connection of the knowledge space with educational material can be based on the use of the ‘Classification’ Category, defined by the IEEE LOM Standard as an element category that describes where a specific learning object falls within a particular classification system.

The integration of IEEE LOM ‘Classification’ Category with ontologies provides a simple way of identifying the domains covered by a learning object.

Since it is assumed that both the domain model and the learning objects themselves use the same ontology, the connection process is then, relatively straightforward.

Figure 3 presents an example of the connection of the two spaces.
The result of the merging of the knowledge space and the media space is a directed acyclic graph (DAG) of learning objects inheriting relations from both spaces.

4. Discovering Optimum Learning Path

In order to extract from the resulting graph of learning objects the “optimum” learning path, we define as optimization criterion the learning time of each learning object.

The learning time of a learning object is defined in the sub-element ‘Typical Learning Time’ of the ‘Educational’ Category, of the IEEE LOM Standard. This sub-element, describes an approximate or typical time it takes to work with or through a specific learning object for a typical intended target audience.

![Diagram of Concept Hierarchy]

Figure 4: Partial View of Concept Hierarchy

\[ G = (V, E) \] is a weighted directed acyclic graph 
\( s \) is the source vertex (starting vertex) 
\( w \) is the weight function (\( w: E \rightarrow R^+ \)) 
\( \text{Adj}[u] \) is the neighbor vertices of \( u \) in adjacency list representation of the graph

For each vertex \( v \in V \), we maintain an attribute \( d[v] \), which is an upper bound on the weight of a shortest path from source \( s \) to \( v \). We call \( d[v] \) a shortest-path estimation. We initialize the shortest-path estimates and predecessors by following \( \Theta(V) \)-time procedure.

**INITIALIZE-SINGLE-SOURCE (G, s)**

1. for each vertex \( v \in V[G] \)
2. \( d[v] \leftarrow \infty \)
3. \( \pi[u] \leftarrow \infty \)

**DAG-SHORTEST-PATHS (G, w, s)**

1. topologically sort the vertices of \( G \)
2. INITIALIZE-SINGLE-SOURCE (G, s)
3. for each vertex \( u \), taken in topologically sorted order
4. \( \text{do for each vertex } v \in \text{Adj}[u] \)
5. \( \text{do RELAX } (u, v, w) \)

Where:
After initialization, \( \pi[\upsilon] = \text{NIL} \) for all \( \upsilon \in V \), \( d[s] = 0 \), and \( d[\upsilon] = \infty \) for \( \upsilon \in V - \{s\} \).

The process of relaxing an edge \((u, \upsilon)\) consists of testing whether we can improve the shortest path to \( \upsilon \) found so far by going through \( u \) and, if so, updating \( d[\upsilon] \) and \( \pi[\upsilon] \). A relaxation step may decrease the value of the shortest-path estimate \( d[\upsilon] \) and update \( \upsilon \)'s predecessor field \( \pi[\upsilon] \). The following code performs a relaxation step on edge \((u, \upsilon)\).

```
RELAX (u, \upsilon, w)
1 if d[\upsilon]>d[u] + w(u, \upsilon)
2 then d[\upsilon] ← d[u] + w(u, \upsilon)
3 \pi[\upsilon] ← u
```

In order to evaluate the total efficiency of the proposed methodology, we have designed an evaluation criterion based on Kendall's Tau [12], which is defined by:

\[
\text{Success} = 100 \times \left( \frac{1}{2} N_{\text{concordant}} - \frac{1}{2} N_{\text{discordant}} \right) \frac{n(n-1)}{n}
\]

where \( N_{\text{concordant}} \) stands for the concordant pairs of learning objects and \( N_{\text{discordant}} \) stands for the discordant pairs when comparing the resulting learning objects ordering with one given by an expert and \( n \) the number of learning object used for testing.

The efficiency of the proposed method was evaluated by comparing the resulting learning objects sequence with those proposed by an expert for 30 different navigation stages (10 cases per concept level) over the concept hierarchy. Average evaluation results are shown in table 2.

It is evident that the proposed method can perform as well as an expert instructional designer.

### 6. Conclusions

In this paper, we address the adaptive learning object sequencing problem in intelligent learning management systems proposing a concrete methodology based on the use of ontologies and learning object metadata. The result is a generic Instructional planner capable of serving for both Adaptive and Dynamic Courseware Generation. The main advantage of this method is that it is fully automatic and can be applied independently of the knowledge domain.

### 7. References


[4]. Brusilovsky P. and Vassileva J., “Course sequencing techniques for large-scale Web-based


