Building intelligent tutoring systems immersed in repositories of e-learning content

Jacek Marciniak *

Adam Mickiewicz University, Faculty of Mathematics and Computer Science, ul. Umultowska 87, 61-614 Poznań, Poland

Abstract

The article presents the principles for creating intelligent tutoring systems, which are active in dynamically growing repositories of e-learning contents. To make such a system work it is necessary to give the contents appropriate structure to enable it to be used in numerous educational contexts. The architecture of intelligent tutoring system is based on wordnet based ontology with expert knowledge, which has been used for repository resource indexing, and which is a basic component of domain model. The solution in its entirety has been supplemented with software agent, responsible for linking the most relevant content in repository to predefined educational strategy.

Keywords: Intelligent tutoring systems; e-learning content repositories; wordnet based ontologies for intelligent tutoring

1. Introduction

A growing number of educational resources created in response to the demands of e-learning opens new possibilities for creation of educational solutions adapted to suit learners’ needs. A dynamic growth of materials results from the fact that they can be created independently by didacticians working with authoring tools and thanks to specifications for interoperability (SCORM, Tin Can). Materials created for one particular training programme can be used in different educational contexts in total or in part. When the materials have a structure that supports reusability it is possible to select from among available components those that fully meet the requirements of learners (custom-made) at the same time realizing pre-set didactic objectives. Contents

* Corresponding author.

E-mail address: jacekmar@amu.edu.pl
adjustment can be done ‘manually’ by the didactician who, after identifying needs, provides the learner with relevant content retrieved from the repository. For massive repositories, however, it is necessary to dispose of solutions that will automatically adjust contents to identified needs, without teacher’s intervention.

The article presents an intelligent tutoring system architecture that adapts to the learner’s needs by delivering contents that is concurrently retrieved from e-learning content repository. The system is nondeterministic as learner while working on materials may receive contents what were not available in the repository when the content designer developed a given pedagogical strategy. Additionally, various e-learning components may be delivered to meet the learner’s needs in keeping with the repository state at a given moment. To make such a system work it is imperative that e-learning content demonstrate adequate structure and that its subject matter has been described with a tool that is general enough to describe contents of diversified subjects. The article presents a method in which wordnet based ontology with expert knowledge is such a tool. This type of ontologies permit describing the subject matter of resources in a way that is analogous to resource tagging on the Internet. Simultaneously, tags are mapped onto concepts due to a basic component of such an ontological system derived from wordnet, namely synset. The pedagogical model of the system is based on pedagogical strategies encoded as SCORM Sequencing and Navigation (SCORM SN) sequencing strategies. This enables creation of educational strategies outside ITS environment as well its modification as the need arises. The solution is supplemented with an ITS Agent which is a software agent responsible for retrieving contents form the repository that fit in with the educational strategy and are in accordance with student model.

The article is organized as follows: Section 2 discusses existing solutions, which combine ITS with e-learning systems. Section 3 discusses the architecture of the system with emphasis on repository structure in which ITS works, the role of wordnet based ontology with expert knowledge and implementation of pedagogical strategies. Section 4 shows what conditions must be met by the system in order to work on massive e-learning content repositories, while Section 5 discusses verification of presented architecture by its implementation in a repository of e-learning content in the area of protection and management of archaeological heritage.

2. Review of existing solutions

Currently existing e-learning solutions are created with technical specifications that guarantee interoperability of didactic content between e-learning platforms (Learning Management System; LMS). The fundamental assumption in this approach is that contents developed by authors are organized in packages (courses, presentations) that exhaustively discuss a subject and are delivered to the learner as a whole. The most common specification is Sharable Content Object Reference Model (SCORM)³, which permits to create e-learning contents mainly in a web-based environment. A newer specification, Tin Can, permits to manage content created also for mobile devices. A part of SCORM specification, which is Content Aggregation Model (CAM) describes the components used in a learning experience, content organization, content packaging, and metadata facilitating storage and exchange. Another part, namely Run-Time Environment (RTE) specifies content launch process in LMS, standardized communication between content and LMS and data model elements for tracking a learner’s experience with the content objects. SCORM 2004 is the most technologically advanced version of SCORM, which introduces additionally Sequencing and Navigation (SN) mechanisms designed to create educational materials that adapt to learner’s needs. SN introduces sequencing concepts (e.g. Activity Trees that represent hierarchical learning activities derived from a content package) and sequencing definition model (control modes, limit conditions and sequencing rules description) that may be used by content developers to define intended sequencing algorithm and behavior within the context of an Activity Tree. A dynamic run-time data model (Tracking model) that capture information gathered from learner’s interaction with the content objects associated with activities, is also defined. As SCORM SN is deemed
complex and difficult by those authors who refuse to get involved with the technological intricacies or to create self-adapting contents, and because of emergence of technological changes, a new specification – Tin Can – aims by contrast to simplify the architecture of the entire solution. Therefore, in light of popularity of SCORM and very good documentation, SCORM SN still remains a tool that in the hands of specialists may be used to build complex educational strategies outside IT systems. Thus, solutions in which such strategies are embedded right in ITS as compiled code can be left behind.

An interesting feature of SCORM 2004 is that in itself it can be treated as ITS. In this particular approach SCORM CAM as well as SCORM SN both play the role of pedagogical model. Student model may be expressed via Tracking Model and SCORM RTE Data model. In this approach, it is assumed that Expert model may be embedded with leaf activities of CAM. Operation of ITS so conceived is always based on the contents collected in content package. Studies on ITS in e-learning environment focused on determination of system architecture of this type. In a solution was adapted, in which the entire architecture was based on Learning Objects generated by the system from available resources. In this system emphasis was laid on domain knowledge management as expressed by ontologies independent of the system. In a solution was presented where ITS is implemented directly onto Learning Object and may be SCORM transferable between LMS systems. However, there are no solutions in which ITS would work on e-learning content repositories that increase dynamically and on which contents are recorded in a specification supporting interoperability.

3. Architecture of ITS active in expanding e-learning content repository

The presented architecture determines the principles of expanding LMS architecture in such a way that it plays the role of ITS on the contents of the entire repository. The solution is based on e-learning content organized with a specification such as SCORM. The entire solution realizes a traditional architecture of ITS system, where components such as Domain model, Pedagogical model and Student model are used to make decisions about subsequent instructional steps. The interface of components of the e-learning contents collected in the repository is acknowledged as tutoring system interface. The system architecture makes it possible to conduct tutoring independently by the system (automatic mode) and to support the didactician by suggesting the content (hand-made recommendation) that is the most relevant at any given stage of instruction (semi-automatic mode). The architecture of the system is presented in Fig. 1.

The system works on content collected in a repository that may be supplemented with e-learning content, e.g. in SCORM. The technical structure of content in repository is compatible with SCORM CAM content organization, i.e. information about content hierarchy, metadata and sequencing behavior is stored. Any number of learning components of SCO or Assets type may be linked to the organization. In the process of content creation content designer (author) defines the location of content on taxonomic path (by ascribing UCTS interpretations), i.e. determines which content may be reused many times. In indexing process subject matter of content is determined by ascribing lexical units from wordnet based ontologies that make up domain model. Lexical units used for indexing make up index of tags.

The system is responsible for sequencing learning components based on the results of a learner’s interactions with the launched content components. This is carried out by using tracking model which captures learner’s interactions with the content components associated with activities from the Activity Tree. Activity Tree is derived from the Content Organization of e-learning course carried by the student. Sequencing is carried out in accordance with the premises of currently realized pedagogical strategy. Pedagogical strategies are mapped onto Activity Tree either from sequencing behavior recorded by content designer in content organization or directly onto the system. In the latter case when creating pedagogical strategies a reference maybe made to a set of pedagogical patterns built in accordance with SCORM sequencing definition model.
As ITS works on the resources of the entire repository and not just on contents of one content package, both pedagogical strategy implemented using sequencing rules in SCORM CAM content structure, and rules implemented on ITS agent are treated as pedagogical model. ITS agent is a software agent, which subject to the realized strategy, can exert diverse influence on learning components delivery. Pedagogical strategy may refer to Student Model that is stored in LMS as independent data structures (Learner data). This approach makes it possible to manage learner’s competence without any limitations, e.g. it is possible to use solutions dedicated to student model management, e.g. such as IMS ePortfolio. The fact that ITS by deciding about consecutive instructional steps goes beyond learning components placed by author in an e-learning course content structure means that the system is responsible for decisions about which sections of materials are the most relevant for a given user at a given stage in the learning process. In making this decision (retrieving module) the following content characteristics must be taken into account: its level of difficulty, target group, duration of course, etc. Such characteristic may be expressed by means of metadata system as for example IEEE Learning Object Metadata (LOM). Usually to describe various types of learning components vocabulary tokens defined for a given metadata element are sufficient (e.g. to describe Resource Type tokens such as exercise, simulation, questionnaire, etc. can be used). However, in order to define the location of content on the taxonomic path in a specific classification system or to define subject matter it is necessary to refer to a more complex description system.

3.1. Content structure in repository

Defining the location of content on taxonomic path determines the structure of contents stored in a repository. This has an impact on how content will be delivered to learner. The task for ITS is to deliver contents in response to identified needs. A question then arises about how large portions of material should be delivered to the learner. It is obvious that the same topic can be exhaustively discussed in a monograph or more perfunctorily in a brief, demonstrative chapter. The difference between such materials does not only result from the method of presenting the material (e.g. considering the needs of target group) but also from the location of a
given component in the context of other components. In the presented architecture the content is described with UCTS (Universal Curricular Taxonomy System). This ensures uniformity of materials in a repository as it has been supplied by various authors over a long period of time.

UCTS is a taxonomic system that makes it possible to describe the location of a content in didactic process without referring to educational context. The system is designed to single out those content structures that guarantee achievement of predefined didactic goals. In UCTS the contents may be structured as follows:

- **Learning Unit (LU)** – smallest content structure that is cohesive in terms of subject matter at the same time making it possible to achieve defined didactic goals;
- **Learning Module (LM)** – a single or numerous LU’s or LM’s. A cohesive portion of content exhaustively discussing a larger issue. LM may contain a component of Exam type (see below) to verify knowledge gained in the module and to modify Learner data;
- **Curriculum** – a complete training program exhaustively discussing a given topic composed of one or more LM’s. The Curriculum may contain an element of the Exam type serving as final exam.

The model assumes that Learning Unit is made up of smaller components that may exhaustively present a (single) topic or verify (defined) competences, but which do not allow to fully realize any didactic goals. Such components are:

- **Learning Object** - a section of material which introduces new contents organized as ‘knowledge capsules’, this element may contain self verification and verification components,
- **Exercise** - a component designed only for self verification of knowledge, the results are not sent to LMS nor to the teacher,
- **Self assessment** - a special type of exercise which enables students to verify their progress in a given section of contents. Questions in this component should have a cross-sectional character,
- **Exam** - a component to verify students’ progress in a defined section of the material; the results should be sent to LMS and made available to tutor,
- **References** - a list of literature to expand on issues discussed.

UCTS enables management of materials that may be retrieved from the repository and delivered to the learner. Those elements are called **Processable Units (PU)**. The choice of materials is determined by the content designer in the process of creating them. It is assumed that smaller components (including Learning objects) may not be delivered to the learner independently as they do not allow to achieve didactic goals that are deemed important in the process of instruction. If they have been retrieved by ITS Agent from repository because they meet the set criteria then they will be delivered to learner along with LU in which they are located. If a Learning Object permits achievement of a single didactic goal important in a given educational context then it should be treated by content developer as Learning Unit (i.e. LU annotation should be attributed to it). This multiplicity of interpretations results from numerous definitions of the term “learning object”.

### 3.2. Resource indexing with wordnet based ontologies

A basic method to describe subject matter of resources (indexing) is tagging it with lexical units taken from wordnet based ontologies. Lexical units derived from such ontologies and used in tagging give the indexer a freedom to choose tags since wordnet is a lexical database. Simultaneously, as lexical units are aligned on synsets there is no possibility of mistakes that might result from polysemy. Reference to wordnet hierarchy (hyponymy, holonymy, antonymy) enables a more precise selection of lexical units in the process of tagging if enhanced tagging system has been used.

Using an ontology whose structure is wordnet-like makes it possible to avoid limitations that may arise while indexing (both manually and automatically) based on terms from controlled vocabularies (thesaurus), or on concepts from domain ontologies. Solutions of this type, as e.g. Getty AAT thesaurus, Cidoc CRM ontology, or even domain ontologies created ad hoc for the needs of a given application, even if very
extensive and carefully prepared, always conceptualize only one domain or sub-domain. With massive content repositories, in which contents may cover a very diversified, and often disparate subject matter, it may prove impossible to use only one system of description. With the assumption that various descriptive systems would be used a problem of the lack of cohesion would appear in resource description. This problem is overcome by the use of wordnet based ontology, in which on the one hand there are no limitations in terms of concepts used in tagging, and, on the other hand, domain knowledge from thesauri or domain ontologies may be mapped onto wordnet structure, which is the backbone of this solution.

3.3. Domain model

A domain model comprises a set of domain ontologies, wordnet based ontology and index of tags. Domain ontologies are to conceptualize a pre-determined domain or sub-domain. This type of ontologies is mapped on wordnet based ontology used in ITS. The mapping is carried out in accordance with the algorithm described in 7. It consists in mapping concepts from ontologies onto synsets, linking them to Domain Categories that were created in accordance with taxonomic structure of ontology and creating the relations between concepts by means of domain relations from wordnet based ontology. Mapping domain models on wordnet-like structure makes it possible to incorporate expert knowledge into ontological system.

Wordnet based ontology is used in pedagogical strategy in the process of constructing queries for the repository. Such queries (Search Conditions) consist of similar concept calculated on the basis of Concept Similarity Rules. The rules are of heuristic character and let determine similarity criteria between concepts e.g. referring to wordnet hierarchy or domain relations between concepts 6.

3.4. Pedagogical model

A pedagogical model consists of Pedagogical strategies, Pedagogical Patterns and ITS Agent. Pedagogical strategy defines how consecutive learning steps are realized by learner. Pedagogical Strategy is implemented by the use of SN sequencing rules. Thanks to this solution it is possible to determine its structure and principles of delivering individual learning components to learners in relation to their progress. The use of SN enables expression of very advanced educational strategies that are being realized. In order to provide learner with learning components retrieved from repository and not only inserted by content designer into content organization SCORM SN was extended. This extension consists in adding a new sequencing control mode (Extension point) devised to point out those learning components in which reference is to be made to repository to find new learning components. Pedagogical strategy may be imported along with content (as package in SCORM 2004) or it may be created in the system based on Pedagogical Patterns.

Pedagogical Patterns is a set of predefined rules which, after mapping onto a given content structure, constitute Pedagogical Strategy. Technically, they are schemes that show how to use SN rules so that a defined system behaviour is achieved in the process of deciding about further instructional steps. Pedagogical Patterns create a catalogue of system behaviours so that Pedagogical Strategies can be created by content designer. Beside reference to the tracking model, which describes possible behaviours of contents included in the content structure by the author at the moment of its creation, Pedagogical Patterns may also refer to Agent ITS so that they suggest repository content to the learner. Typical pedagogical patterns include for instance Case Study Inclusion Strategy Pattern, which enables retrieval from repository of case studies which complement given theoretical material, or Similar Content Strategy Pattern, which is responsible for searching repository for the most adequate content in response to an identified competence gap. The last rule (pseudocode) may look like this:

If a competence gap has been detected for the topic described by the concept X, then insert after the current item the item retrieved from the repository that meets the Search Conditions.
The Code for the rule above is presented in Figure 2.

```xml
<manifest>
  ...
  <title>Knowledge test</title>
  <imsmanifest:sequencing>
    <imsmanifest:sequencingrules>
      <imsmanifest:preConditionRule>
        <imsmanifest:ruleCondition condition = "completed"/>
        <imsmanifest:ruleCondition condition = "skip"/>
        <imsmanifest:ruleCondition condition = "true"/>
      </imsmanifest:preConditionRule>
    </imsmanifest:sequencingrules>
    <imsmanifest:objectives>
      <imsmanifest:objective objectiveID = "its.run.strategy.SimilarContentStrategy"/>
    </imsmanifest:objectives>
  </imsmanifest:sequencing>
  ...
</manifest>
```

```java
if (cmi.interactions.objective == "incorrect") {
    getAPI()[1].setValue("cmi.completion_status", "incomplete");
    getAPI()[1].setValue("cmi.success_status", "failed");
}

if (cmi.objectives[0].id == "its.run.strategy.SimilarContentStrategy") {
    searchConditions = cmi.interactions.objective + "+UTCS=PU+";
    getAPI()[1].setValue("cmi.objectives[0].description", searchConditions);
}
```

Fig. 2. Implementation of Similar Content Strategy rule

ITS Agent is a software agent in charge of retrieving content from repository (Figure 3). In the LMS is embedded as an autonomous software component that realizes a defined strategy. This strategy may refer only to simple changes occurring in the environment (the environment here being Tracking Model on which Agent works), or it may refer to knowledge stored in Student Model (e.g. it is possible to retrieve the information that a student prefers studying with materials that contain video sequences). Since the agent operates independently on the course Activity Tree, Trigger conditions must be met for it to take action. Those conditions may invoke ITS Agent to take action for all the content components of a defined type (e.g. all elements of the type UCTS Exam), or for a single element where it is necessary to modify tree of the course. Trigger conditions are transferred to ITS Agent only in Extension Point by means of the mechanism of objectives.

![Diagram of ITS Agent](image)

Fig. 3. Content retrieving

An example algorithm of simple reflex ITS Agent is presented below.

```java
if TriggerConditions {
  searchForContentInRepository(SearchConditions);//retrieving
  findBestCandidateFromRetrivedContent();// recommending
  insertBestCandidate(PlaceToInsert);// inserting
}
```
ITS operation varies depending whether the system is working in automatic or semi-automatic mode. In the semi-automatic mode the teacher himself decides (hand-made recommendation) which content retrieved from repository that meets the set criteria is to be delivered to the learner. In the automatic mode it is the ITS that makes such a decision with reference to recommendation rules (e.g. Case Study Inclusion Recommendation Rule). Recommendation Rules are heuristics whose task is to select best suited content for a given pedagogical strategy. ITS operation is also correlated to the method of constructing criteria responsible for retrieving content from repository (SearchConditions). The level of enquiry specificity may vary and may depend on the author of the pedagogical strategy under construction. ITS may serve simple searches with reference to content structure and metadata that describe it (e.g. taxonPath = UCTS:LM, tag = “Altamira”, difficult = easy). It may also refer to a relation with wordnet based ontology that creates a Domain model thus making it possible to search for similar concepts (i.e. tag="Altamira:similar:1"). As a result of question thus asked those learning components will be retrieved from repository that suit the learner’s needs to a varying degree (search precision is correlated to the types of relations that have been expressed in domain model). Therefore the final decision about which learning component is to be delivered to the learner is made by the teacher (semi-automatic mode) or by recommendation strategy (automatic mode).

4. Model in action

To make the system work it is necessary to adequately prepare didactic contents stored in the repository. They must be divided into PU and annotated with metadata including metadata describing subject matter. Dividing into PU’s should be carried out by the content designer. While working on the content structure the author should decide which parts of the material after being removed from the initial context may function independently in other educational contexts. Typically, these decisions will affect the technical structure of the content, i.e. PU’s must have a structure that will allow their technical isolation from materials in which they were originally immersed. How detailed the metadata description of contents will be depends on Pedagogical Strategies and on how much detailed information about the learner is stored in the Student Model. First of all metadata should be ascribed to those content components that have been marked as PU’s. It is also possible to ascribe metadata to more granular content components (e.g. Learning objects, or assets). Whether such information will be used depends on the fact if such behavior has been included in recommendation strategies. Tagging of resources subject matter should be carried out by the authors themselves, without any limitations as to lexical units used. If in the wordnet based ontology it appears that certain vocabulary necessary for tagging is missing it should be added by means of an algorithm, described in 7. In the process of tagging beside lexical units also information about the version of ontology and sense lexical units e.g. pmah:archaeological heritage:1 should be added.

Decision making by Pedagogical Model that refers to resource subject matter requires that the same description method is used to describe the resource and in the Pedagogical patterns implementation. This means that Learner’s interactions with launched content must be described with concepts from the same ontological system that has been used to describe subject matter stored in the repository. In the presented approach this system of description is made up of concepts from wordnet based ontology. From a technical standpoint in this process these concepts are recorded as interaction objectives (cmi.interactions.n.objective) linked to e.g. questions in quizzes that are used in the process of conditional sequencing of activities at SN level (Figure 2). They are then mapped onto objectives (cmi.objectives) and transferred to ITS Agent at Extension Points as criteria for searching repository.

An important assumption of the presented ITS architecture is that Pedagogical Model is implemented with mechanisms (CAM and SN) that allows to create pedagogical strategies outside ITS, directly on the content. Thus, it is possible to upload content into LMS and just launch given pedagogical strategy. It is also possible to map onto content stored in the repository a chosen pedagogical pattern directly in the system. In the latter case,
content in SCORM 2004 must be deprived of sequencing rules. In the case of other specifications (SCORM 1.2, TinCan, AICC) content organization must be mapped onto organization of SCORM 2004.

5. Applications

The presented architecture was implemented for the needs of a system that manages e-learning contents in the area of protection and management of archaeological heritage used to organize distance training courses addressed at various target groups, with varying complexity, duration and level of difficulty. This repository covers multimedia interactive contents in SCORM with approximately 4800 SCO that make up about 800 UCTS LU, 220 UCTS LM and 23 UCTS Curricula. The contents is in five languages: English, German, Latvian, Polish and Spanish. The division into UCTS LU and LM was made by the authors of contents. The curricula were built in an external tool - Content Repository Tool – directly by trainers who prepared materials for distance training courses. In this process a need was observed in terms of tools to support the trainer who works on massive content repositories. For example, a need emerged to extend materials with components from the repository in a situation when deficiencies were detected in learner’s knowledge. Another interesting problem was the necessity to supplement contents of a general character with case studies from the learner’s own culture (e.g. Spaniards expected case studies from the area of Mediterranean, or Swedes - from Scandinavia).

The above-mentioned needs were achieved in the architecture ITS Agent that realizes simple reflex agent described above and pedagogical strategies built with two Pedagogical Patterns and then mapped onto content with detail level of UCTS Module. The entire solution is based on content that was tagged by authors by means of wordnet based ontology with embedded domain conceptualization involving content structure in the repository (PMAH Ontology). The ontology is made up of 150 Domain Categories and 1500 Lexical Units. ITS Agent was implemented in the Edumatic system, which is a system of LMS/LCMS class with full implementation of SCORM SN.

The usefulness of the system architecture was verified by implementing two sample pedagogical strategies that were tested on a technological platform presented above. To implement Pedagogical strategies two e-learning courses in SCORM were used as well as two pedagogical patterns (Case Study Inclusion Strategy Pattern and Similar Content Strategy Pattern). Pedagogical strategies were implemented by mapping pedagogical patterns onto content structure of e-learning courses. The mapping was carried out on Extension Points that were UCTS Exam-type learning components designed to verify learner’s progress. Trigger points were implemented onto learning components that make it possible to set off ITS Agent. Case Study Inclusion Strategy was tested on the course ‘Mentalities and perspectives in archaeological heritage management’ (course components complexity: 1 UCTS LM, 3 UCTS LU and 28 SCO). In the repository there were related case studies about archaeological sites (Altamira in Spain and Heathrow Terminal 5 in England). In automatic mode recommendations were realized by heuristics implemented in Use Case Inclusion Recommendation Rule. The heuristic assumes that learner must first be provided with case study about his native country. This premise was checked by comparing tags describing content with data about learner’s native country collected in Learner data. The other strategy was verified on the course ‘Introduction to archaeology” (1 UCTS LM, 6 UCTS LU, 37 SCO). In this course decisions about what content is to be retrieved from repository are made in the extension point, based on test results.

In the tests that were carried out so far the system operated correctly, i.e. it retrieved contents from repository that were adequate to learner’s needs as identified in tests in accordance with criteria determined by content designer. In the semi-automatic mode it enabled the didactician to quickly decide which content selected from the repository should be delivered to the learner. Hence it is possible to individualize the training of large student groups with mixed competence. In the automatic mode the system provided the learner with content that was in accordance with content designer brief without the need to wait for teacher’s decision. In
either case the system rationalized the process of selecting content from a repository that the trainer is not fully familiar with its list of contents.

6. Conclusions

In the article a solution is presented that makes it possible to realize ITS active in e-learning contents repositories. In massive repositories, a mechanism to support the didactician in the organization of didactic process is necessary; the same goes for solutions that will enable a fully automatic use of repository resources with individualized learning.

The presented architecture is based on SCORM SN, or a technology that is widespread, mature and stable. The presented idea to create pedagogical strategies based on pedagogical patterns enables didacticians to simply and conveniently create didactic content that adapts to learners’ needs, at the same time shifting responsibility for creation of pedagogical patterns to content authoring professionals, who create a library of ready-made solutions for the former.

A crucial point of the architecture is immersion in the LMS core of a so-called ITS Agent, or software agent that enables extension of contents in e-learning courses with contents stored in a repository. The article discusses the principles of using the most simple agent of this type, i.e. Simple Reflex Agent. Further studies are planned on more developed agents, especially on agents referring to information on student progression collected for longer periods of time in the Student Model. Studies are also planned on other pedagogical strategies with particular emphasis on situations in which subject matter of learning components is not fully described in the repository.

References