A Mobile and Adaptive Language Learning Environment based on Linked Data

Davy Van Deursen[1], Igor Jacques[2],
Stefan De Wannemacker[2], Steven Torrelle[1], Wim Van Lancker[1],
Maribel Montero Perez[2], Erik Mannens[1], and Rik Van de Walle[1]

[1] Ghent University - IBBT,
Gaston Crommenlaan 8/201, B-9050 Ledeberg-Ghent, Belgium
`firstname.lastname@ugent.be`

[2] ITEC - Interdisciplinary research on Technology, Education and Communication,
K.U. Leuven Campus Kortrijk, Etienne Sabbelaan 53, B-8500 Kortrijk, Belgium
`firstname.lastname@kuleuven-kortrijk.be`

Abstract. The possibilities within e-learning environments increased dramatically the last couple of years. They are more and more deployed on the Web, allow various types of tasks and fine-grained feedback, and they can make use of audiovisual material. On the other hand, we are confronted with an increasing heterogeneity in terms of end-user devices (smartphones, tablet PCs, etc.) that are able to render advanced Web-based applications and consume multimedia content. Therefore, the major contribution of this paper is an adaptive, Web-based e-learning environment that is able to provide rich, personalized e-learning experiences to a wide range of devices. We discuss the global architecture and data models, as well as how the integration with media delivery can be realized. Further, we give a detailed description of a reasoner, which is responsible for the adaptive selection of learning items, based on the usage environment and the user profile.

Keywords: Adaptive, Language Learning, Mobile, Web-based

1 Introduction

The last years, the use of e-learning environments has increased spectacularly, not only in formal educational settings, but also in working and private environments. At the same time, the possibilities within these e-learning environments increased dramatically: learning environments for instance have become easier and more pleasant to use, they allow various types of tasks and fine-grained feedback, and they can make use of audiovisual material. Moreover, while e-learning environments were traditionally offered as applications on stand-alone computers, nowadays they are more and more being rendered over the Internet. It is clear that these evolutions are related to technological evolutions, and the wide availability of fast multimedia computers and internet access.

Next to the fact that e-learning environments are more and more deployed over the Web, we are confronted with an increasing heterogeneity in terms of end-user devices that are able to connect to the Web and consume multimedia content. Therefore, personal devices such as tablet PCs and smartphones could be used as learning devices, next to traditional desktop and laptop devices. Also, the role of personalization within e-learning environments has become more and more important. Personalization can be applied both at the learning level (i.e., adjust learning sessions according to the learner's capabilities) and at the environmental level (i.e., adjust the rendering of the learning environment according to the characteristics of the usage environment).

The above described challenges are exactly the ones that are currently tackled in the IBBT MAPLE project (Mobile, Adaptive & Personalized Learning Experience[3]), which aims to make adaptive mobile e-learning possible. Therefore, in this paper, we present a Web-enabled e-learning environment that is able to offer personalized learning sessions on any device, primarily focused on language learning making optimal use of digital multimedia. In order to realize such an environment, we need the following key components:

– a *common, machine-understandable data model* that is independent of usage environments and is able to express both learning content and metadata about the learning content;
– a *logging framework* that allows to capture the behaviour and performance of the learner on a detailed level;
– a *reasoner* that is able to select learning items based on the learner's capabilities and behaviour;
– a *media delivery platform* taking into account usage environment characteristics and restrictions.

In the remainder of this paper, we provide an overview of the architecture of our adaptive e-learning platform. Further, we discuss the above described key components in more detail. Finally, we discuss related work, draw a number of conclusions, and discuss some future work.

2 MAPLE platform

In order to offer a highly adaptive e-learning platform that can also deal with (mobile) multimedia delivery, we designed the architecture that is depicted in Fig. 1. Two major parts can be distinguished: the e-learning platform and the media delivery platform. The e-learning platform relies on two RDF stores, i.e., a store for learning exercises and a store for learner profiles. The learning items store is filled through the learning item ingest service. More details regarding the creation of learning items and the data model according to which they are modeled are provided in Section 3. Further, the learner profile store is build up, based on the learners' actions and preferences (see Section 3.5). The reasoner is responsible for selecting the most adequate exercise, based on the learner's profile and environment and the available learning items. Detailed information

[3] `http://www.ibbt.be/en/projects/overview-projects/p/detail/maple-2`

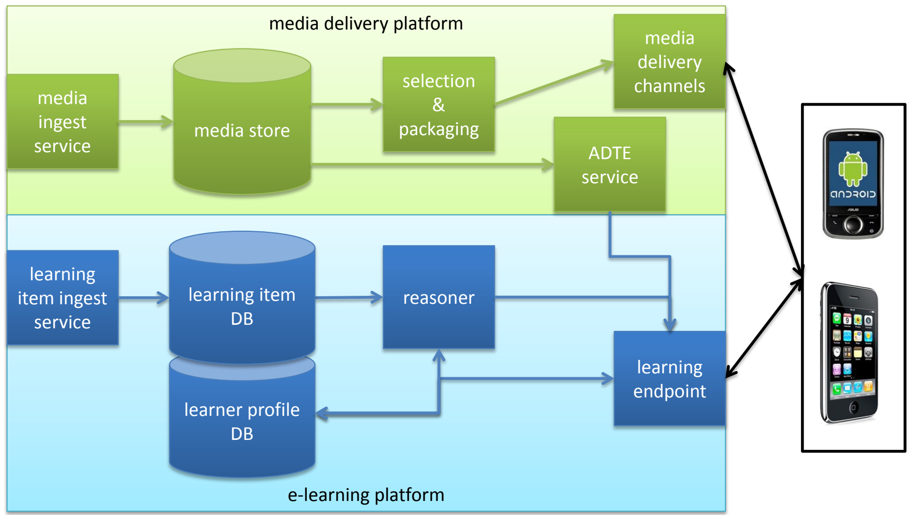

Fig. 1. The MAPLE e-learning platform

regarding the reasoner is provided in Section 4. Finally, the learning endpoint is the communication point between learner devices and the e-learning platform.

The media delivery platform corresponds to NinSuna[4], which is a metadata-driven media adaptation and delivery platform [25]. At its core, format-independent modules for temporal selection and packaging of media content are present. Almost all existing media delivery channels are supported by NinSuna: RTSP, RTMP, HTTP progressive download, and HTTP adaptive streaming. Moreover, native support for Media Fragments 1.0 [24] is provided, which enables the delivery of media fragments (i.e., temporal or track fragments) in a standardized way [15]. Finally, NinSuna comes with an Adaptation Decision-Taking Engine (ADTE), which is able to 1) detect the capabilities of the device issuing the request and 2) take a decision regarding which quality version of the requested media resource is the most adequate for the detected device. A more detailed description of the NinSuna platform can be found in [25].

The presented e-learning platform exposes its data (i.e., learning content and accompanying media resources) as linked data. More specifically, it follows the guidelines regarding the publication of linked data[5]: use dereferencable HTTP URIs as names for things, provide useful information using the standards (RDF, SPARQL), and include links to other URIs. Hence, within our platform, the learning items and learner profiles are available through a SPARQL endpoint, while the metadata of the media resources are published as RDF URIs. This way, services such as the reasoner and the ADTE can rely on the linked data and can start reasoning over it.

---

[4] `http://ninsuna.elis.ugent.be`

[5] `http://www.w3.org/DesignIssues/LinkedData.html`

A typical e-learning scenario using this architecture is then as follows:

(1) the learner logs in into the Web-based e-learning application using its mobile device, which contacts the learning endpoint of the e-learning platform; the end point approaches the reasoner which provides a personalized overview of the available courses;
(2) based on the course selected by the learner, the reasoner selects an exercise from the learning item store, taking into account the learner profile and the available exercises within that course;
(3) when the selected exercise contains media content (audio, video, or images), the ADTE of NinSuna is contacted in order to select the media resource version that fits best for the current device;
(4) the learning endpoint renders the selected exercise in HTML and sends the response to the learner;
(5) when the learner is solving the selected exercises, his/her answers and his/her behaviour in terms of clicks and timing is logged and sent back to the e-learning platform;
(6) the received answers and behaviour information are used to update the learner's profile.

In the next sections, more detailed information regarding a number of components in the architecture is provided.

3 Data Models and Instance Generation

A number of different data models need to be developed in order to structure and define the content used on the e-learning platform. More specifically, we need the following data models:

- model for the learning items and their metadata (e.g., question, possible answers, difficulty level);
- model for the learning domain;
- model for the metadata of the media resources (e.g., bit rate);
- model for the learner profile;
- model for the logging.

In the following subsections, we provide more information regarding these different models and how they are populated. Note that all ontologies are modelled in OWL and published online.

3.1 Model for learning items and their metadata

The model for learning items consists of two ontologies: one for the learning items themselves[6] and one for their metadata[7]. An example instance of a learning

[6] `http://multimedialab.elis.ugent.be/organon/ontologies/maple/content`
[7] `http://multimedialab.elis.ugent.be/organon/ontologies/maple/llomp`

Listing 1.1. Representing a learning item and its metadata in RDF (in Turtle).

```
@prefix mplc: <http://multimedialab.elis.ugent.be/organon/ontologies/
     maple/content#>.
@prefix llomp: <http://multimedialab.elis.ugent.be/organon/ontologies/
     maple/llomp#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dc: <http://purl.org/dc/terms/> .

<http://ninsuna.elis.ugent.be/rdf/resource/maple/blcc_47363>
  a llomp:Exercise ;
  dc:title "47363" ;
  mplc:exerciseType mplc:MultipleChoice ;
  mplc:media <http://ninsuna.elis.ugent.be/Media/Maple/FLAA2V0#t=0,19> ;
  mplc:task "What do Belgians eat, according to the reporter?"@en,
        "Wat eten de Belgen volgens de reporter?"@nl ;
  mplc:answerSpace "Les Belges mangent ..." ;
  mplc:input [
    a mplc:Input ;
    mplc:answer [
      a mplc:Choice ;
      mplc:isCorrect "false"^^xsd:boolean ;
      mplc:scoreCorrect "0"^^xsd:int ;
      mplc:scoreFalse "0"^^xsd:int ;
      mplc:text "des frites"@fr .
    ] ;
    mplc:answer [
      a mplc:Choice ;
      mplc:isCorrect "true"^^xsd:boolean ;
      mplc:scoreCorrect "1"^^xsd:int ;
      mplc:scoreFalse "0"^^xsd:int ;
      mplc:text "de la glace"@fr .
    ] .
  ] ;
  mplc:maxScore "1" ;
  mplc:minScore "0" ;

  llomp:educational [
    a llomp:Educational ;
    llomp:difficulty llomp:medium ;
    llomp:level llomp:A2 ;
    llomp:learningComponent :learningComponent_44854 .
  ] ;
  llomp:lifeCycle :lifeCycle_47363 .
```

item modelled according to our model is shown in Listing 1.1. We explain and illustrate both ontologies based on this example. The model is heavily based on the Learning Object Metadata (LOM, [2]). LOM specifies a conceptual data scheme and the corresponding XML-binding for metadata of learning items. We started from LOM and defined a number of extensions in order to provide improved support for learning subject, feedback and scoring, as well as better integration with media resources. Further, as mentioned before, we split our model between learning items and their metadata.

We describe not only the metadata of learning items, but also the exercises themselves. This way, they are formally represented, independent of any rendering. Moreover, they can be easily integrated with their metadata and corresponding media resources. Also, the reasoner (Section 4) will not only rely on the learning item metadata, but also on the items themselves (e.g., this type of exercise is preferred by the learner). For the moment, six `mplc:exerciseType`s are supported (focussed on language learning):

– Multiple Choice: given a number of answers, the learner has to choose exactly one answer;
– Multiple Response: given a number of answers, the learner has to choose one or more answers;
– Fill Gaps: given a text with some gaps, the learner needs to fill in missing text in text boxes;
– Dropdown: same as Fill Gaps, but instead of free text fields, the learner can choose between a number of predefined answers;
– Click on Text: given a text, the learner needs to click/tab on one or more words;
– Click on Zone: given an image or video, the learner needs to click/tab at one or more regions within the image or video.

Note that media elements can also occur within the first five types of exercises. For instance, a movie can be played followed by the question to solve. Only the last type (Click on Zone) uses multimedia in an interactive way as described in [19].

In Listing 1.1, a multiple choice exercise is used as example (line 9). A link to a movie fragment is provided via the `mplc:media` property (line 10), which takes as value a Media Fragment URI (see Section 3.3). The `mplc:task` description (line 11) provides the question or task in multiple languages (based on the level of the learner, the reasoner can choose if the language is presented in the native language of the learner or not). Further, the `mplc:answerSpace` (line 13) corresponds to the zone where the learner can enter its answers. Within such an answer space, `mplc:input` is provided (line 14), where each `mplc:answer` corresponds to one possible answer. In case of a multiple choice type, each answer corresponds to a `mplc:Choice`. It contains information such as 'is this possible answer the correct one?', 'how much does the learner score when (s)he selects this one?', and the possible answer itself. LOM-specific elements such as `llomp:lifeCycle` (line 40) and `llomp:educational` (line 34) are present as well.

As a part of the aforementioned LOM extensions, we added the learning component property to the educational component. Since the MAPLE project focusses on language learning, we extended this learning component property with specific support for language learning. The learning component is split up into three separate subcomponents: target language, theme and language component. The latter component can have one or more of the following subproperties:

– knowledge property: vocabulary, pronunciation, etc.;
– skill property: reading, listening, writing or speaking

We also defined a hierarchical structure for the range of the knowledge property based on which the exact knowledge URIs can be deduced. This was done in a language-independent way extendable with language-specific elements. As the

Listing 1.2. Representing a learning component in RDF (in Turtle).

```
@prefix lang: <http://kuleuven-kortrijk.be/itec/ext/ontologies/
    itec_elearning_ontology/languagecomponent/#>.
@prefix llomp: <http://multimedialab.elis.ugent.be/organon/ontologies/
    maple/llomp#>.

<http://ninsuna.elis.ugent.be/rdf/resource/maple/learningComponent_40001>
  a llomp:LearningComponent ;
  llomp:theme "agriculture" ;
  llomp:targetLanguage "en-UK" ;
  llomp:languageComponent [
    a llomp:LanguageComponent ;
    llomp:knowledge <http://kuleuven-kortrijk.be/itec/ext/ontologies/
        itec_elearning_ontology/languagecomponent/grammar/partsOfSpeech/
        substantive> ;
    llomp:skill lang:writing .
  ] .
```

skill and knowledge property exists next to each other, it is possible to specify the subject of an exercise very accurately. In Listing 1.2 an example instance of a learning component can be found. The exercise in this instance trains the writing skill of substantives related to agriculture.

Within the MAPLE project, we use learning items from Televic Education (TEDU)[8]. Currently, TEDU stores their learning items and accompanying metadata in a SQL store. Through XML feeds, the store can be accessed from outside. Hence, we implemented a converter taking as input the XML feeds and producing RDF learning items according to the above described model.

3.2 Model for the learning domain

The learning items are not physically arranged into courses. Which learning objects belong together is determined by the metadata, namely the learning component within the educational component of each item. The domain model consists of two type of relations: prerequisite and hierarchical relations. In the project, the domain model is supposed to be simple. It is a three level hierarchical model in which the items are first distinguished by their target language, secondly by their theme, and thirdly by their language component. Additionally, there exist prerequisite requirements between the language components, expressing one language component depends on the knowledge of another. The reasoner will take into account these prerequisites when determining what courses are available for the learner.

3.3 Model for media metadata

To model media resources, we rely on the W3C Media Annotations ontology [11], which is supposed to foster the interoperability among various kinds of metadata formats currently used to describe media resources on the Web. Moreover, it

[8] http://www.televic-education.com/en/

Listing 1.3. Representing a learner profile in RDF (in Turtle).

```
@prefix itec: <http://kuleuven-kortrijk.be/itec/ext/ontologies/
    itec_elearning_ontology#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix mplc: <http://multimedialab.elis.ugent.be/organon/ontologies/
    maple/content#>.

<http://kuleuven-kortrijk.be/itec/ext/ontologies/itec_elearning_ontology/
    maple/learners#blcc_piet_lambrecht>
  foaf:nick "piet_lambrecht" ;
  foaf:firstName "Piet" ;
  foaf:lastName "Lambrecht" ;
  itec:hasProficiency [
    a itec:Proficiency ;
    itec:hasLearningSubject :learningComponent_47584 ;
    itec:hasScoredEvaluation [
      a itec:ScoredEvaluation ;
      itec:score "3.2"^^xsd:float ;
      itec:scoreVariance "1.1"^^xsd:float ;
      itec:scoreScale itec:defaultEuropeanLanguageLevelScale .
    ] .
  ] ;
  itec:hasLearningGoal [
    a itec:ScoredEvaluationLearningGoal ;
    itec:hasScoredEvaluation [
      a itec:ScoredEvaluation ;
      itec:score "4"^^xsd:float ;
      itec:scoreScale itec:defaultEuropeanLanguageLevelScale .
    ] ;
    itec:hasLearningSubject :learningComponent_47584 .
  ] ;
  itec:preferredExerciseType mplc:DropDown .
```

already contains mappings to many other existing metadata formats. Further, the ontology also provides support for Media Fragment URIs.

3.4 Model for the learner profile

In order to steer the decision making of the reasoner, an up-to-date learner profile is required for each of the learners in the learning system. This profile holds proficiency score estimations for each of the appropriate learning subjects. Each of these values is supplemented with a reliability parameter, namely the variance of the estimator. As we focus on language learning, the proficiency scores are expressed on a continuous scale based on the discrete European Language Levels [4]. The level of A1 conforms to a score of 0, A2 to 1, B1 to 2, etc. Also, the profile keeps a list of the learning goals which were set for that learner. An example of such a learning goal could be “Achieve the B2 level for the French verb form imparfait”. The type of learning items the learner prefers can also be saved in the profile. An example instance can be found in Listing 1.3.

The properties in the model will be caught either automatically either by means of preference setting. The learner’s favourite learning item types can be edited through a preference menu and the learner’s proficiency scores will be updated by a module of the reasoner. Additionally, the ontological model sup-

Listing 1.4. Representing a logging abstract in RDF (in Turtle).

```
@prefix itec: <http://kuleuven-kortrijk.be/itec/ext/ontologies/
    itec_elearning_ontology#>.
@prefix learners: <http://kuleuven-kortrijk.be/itec/instances/maple/
    learners#>.
@prefix log: <http://kuleuven-kortrijk.be/itec/instances/maple/logging#>.
@prefix maple: <http://ninsuna.elis.ugent.be/rdf/resource/maple/>.

log:learnerSession12452
  a itec:LearnerSession ;
  itec:hasSessionStart "2010-10-26T21:32:52.126"^^xsd:dateTime ;
  itec:hasSessionStop "2010-10-26T21:38:52.526"^^xsd:dateTime ;
  itec:hasLearner learners:blcc_piet_lambrecht ;
  itec:hasSubSession [
    a itec:LearningSession ;
    itec:hasSessionStart "2010-10-26T21:32:52.229"^^xsd:dateTime ;
    itec:hasSessionStop "2010-10-26T21:38:52.501"^^xsd:dateTime ;
    itec:hasItemObjectSession [
      a itec:ItemObjectSession ;
      itec:hasItemObject maple:blcc_47363 ;
      itec:hasSessionStart "2010-10-26T21:32:56.233"^^xsd:dateTime ;
      itec:hasSessionStop "2010-10-26T21:32:59.999"^^xsd:dateTime ;
      itec:hasAnswerSubmittedEvent [
        itec:hasInputObject maple:inputObject_57495 ;
        itec:hasGivenAnswer maple:answer_57495 ;
        itec:dateTime "2010-10-26T21:32:59.526"^^xsd:dateTime .
      ] .
    ] .
  ] .
```

ports properties like motivation, learning style, learner strategy, and cognitive ability's, but currently these are not used in the MAPLE e-learning platform.

3.5 Model for logging the learner's activity

Finally, we developed a model for describing logging information. For instance, the model is able to express information such as the start and stop of a learner session or the learner's course selection. Once the learner has chosen a course, a learning session is initiated in which the reasoner successively selects a new learning item, each time resulting in a learning item session which lasts for the time the learner interacts with the item. During such an item session a learner can give an answer, request a hint, or change his mind by changing his answer. All these interactions are logged by the system. This results in a huge amount of information which is consumed in two ways. Firstly, a part of the logging information is used at run-time by the reasoner. For instance, a score attained by the learner will affect the proficiency score of a learner's profile through the functionality of the reasoner's proficiency manager. Secondly, after runtime, the logged information will be used as input for statistical research tracing how certain interactions of the learner give information about the learning process. In Listing 1.4 an example instance can be found. The learner and learning session, and the session of the item are respectively interconnected by the `itec:hasSubSession` and the `itec:hasItemObjectSession` relation.

These resulting triples are partially generated in the core of the reasoner, e.g. the start en stop of the learner and the learning sessions. The low level interactions concerning one specific exercise are generated at the client and sent back to the reasoner which processes the logging and stores it in the learner profile RDF store.

4 Adaptive Learning Item Selection

The reasoner, introduced in Section 2, is a crucial component within the MAPLE learning system architecture as it is responsible for the adaptive learning item selection. If a learner logs in, the reasoner will first of all provide a short list of courses from which the learner can choose. As the reasoner is aware of the learning goals for each learner through the learner profile model, only courses that contribute to the not yet attained learning goals can be selected. Next, once the learner has chosen a course, the reasoner will start up a learning session and will successively decide on the exact exercise to deliver to the learner.

The reasoner takes into account the learner profile as well as some real time environmental properties. For the environmental adaptivity, both the screen capacity and connection quality of the user's device are sources of adaptivity. In case the screen size is too small, the reasoner will avoid the use of exercises with media. A slow network connection will also result in avoiding media exercises. For the learner profile adaptivity, there are two main policies which can steer the decision process. The first one is based on a theory stating that the exercise difficulty needs to be increased each time a learner has answered a series of four exercises correctly. Similarly, when four consecutive exercises are answered incorrectly, it should go down [12]. The second policy is based on a pedagogical theory which tries to keep the learner's motivation high by chasing a predefined (e.g. 70 %) correct-answer probability. This probability can be estimated based on the IRT theory ([5]) by combining the current proficiency estimation with the level and difficulty of the exercise [28, 7]. The aforementioned policies are supplemented with an event-driven feedback system. The system allows the sequencer to shift in a feedback item (instead of an exercise) to explain a learning subject once a specific and predefined condition is met. For instance, "the learner made five errors in a row against the same learning subject". This feedback item is chosen based on the learning component property which both the feedback and the exercise item have in their metadata. For both policies, also the preferred exercise types of the learner are taken into account by favouring them though not completely cold-shouldering the other exercise types.

To fulfil the aforementioned tasks, the architecture of the reasoner (shown in Fig. 2) consists of six modules, supplemented by a facade for communicating with the learning endpoint. The six reasoner modules are the Learner manager, Environment manager, Learning task decision manager, Sequence manager, Logging manager and Proficiency manager. We elucidate the functionality of these modules by means of the following example.

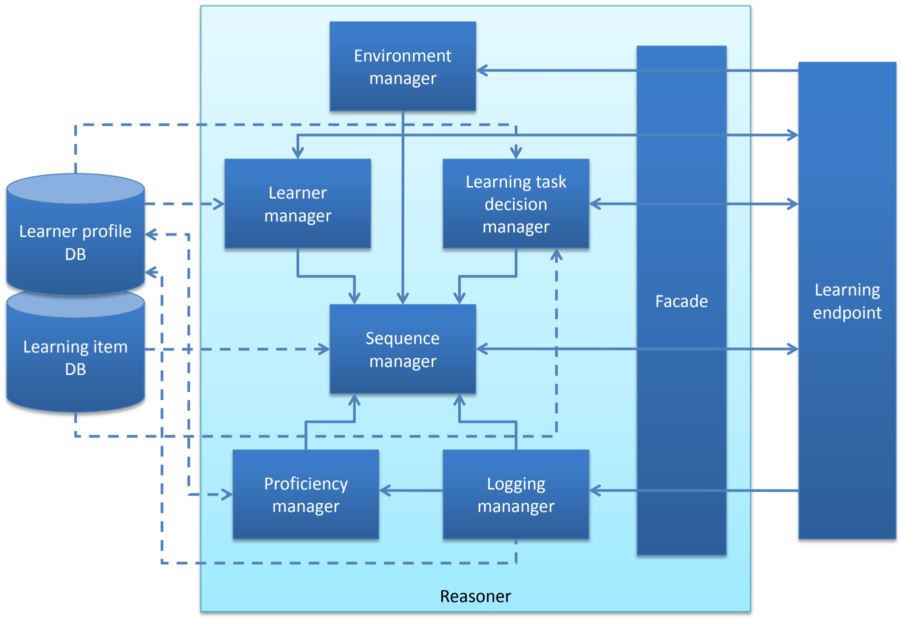

Fig. 2. The reasoner architecture

Suppose a learner's initial profile was set by a teacher thereby providing the learning goal "Achieve the B2 level for the French verb form imparfait" and also providing an estimation for the learner's initial level, namely A2, for "the French verb form imparfait". When the learner logs in, the Learner manager produces a learner session. Consequently, the Learning task decision manager loads the learner's learning goals in order to compose a three-level tree representation of all courses relevant for this learner, as explained in Section 3.2. This tree is sent to the learning endpoint which produces a representation such that the learner can navigate through the tree. Let us assume that the learner first selects 'French' followed by the theme 'General' and finally the language component 'Imparfait'. Besides, the learner opens the preferences menu and sets the dropdown exercise type as his favourite one.

Next, the Learning task decision manager composes a learning task object which is send to the Sequence manager. Here the learning task is sequencing the items (exercises and feedback) with the first policy of adaptivity, starting from level A2, having as a stop criterion the achievement of the level B2, and taking into account the learner's preferred exercise types and environmental properties. Subsequently, the Sequence manager loads the sequencer necessary for the learning task. To this end, the sequencer makes use of the Environment manager, which is an access point for information on the current connection quality and the screen size of the device of the learner. At this point, the sequencer can successively decide on the id of the next item and passes its choice to the learning endpoint, which automatically generates a visual representation and makes use of the delivery platform in case media are present.

Once the learner finishes the exercise or has read the feedback in case of a feedback item, the logging information about the interactions of the learner with the item are sent back to the Logging manager of the reasoner. The latter sends this information as a specific logging object to a couple of observer objects which all have different functionalities. For instance, there is an observer writing these logs to the learner profile RDF store. Another observer warns the sequencer when for example four exercises have been consecutively answered correctly and yet another sends the learner's score to the Proficiency manager together with the level, difficulty and the learning subject of the answered exercise. The Proficiency manager keeps the proficiency scores up to date. Prior to every decision of the sequencer, the stop criterion is tested based on a proficiency that is retrieved from the Proficiency manager. If this criterion is reached, the sequencer sequences a special concluding feedback item announcing the end of the learning session to the learner.

5 Related Work

The architecture of the reasoner builds further on existing proposals for generic learning system architectures such as in [20]. These architectures however have mostly been designed having an adaptive hypermedia learning system in mind. Even though most systems currently developed are based on providing learner control based on adaptive links, e.g. [3], our system is specialized in adaptive curriculum sequencing, meaning that the learning objects are sequenced in an automated way. To create an adaptive learning system the method of using ontologies has often been proposed in literature, e.g., in [23, 17, 8]. We partially rely on existing ontologies and data models, and introduced new data models such as a model for describing learning exercises and language-learning specific information. The latter were all done in collaboration with educationalists. Additionally, both the delivery platform and the reasoner take into account connection quality and screen size either to choose the right video format either to avoid sending any media to a device if they cannot be delivered in an optimal way. This way, our system implements a part of the context-awareness which has been claimed to be crucial in mobile learning [23, 27].

The ontology for the learner profile is a compact non-exhaustive synopsis of the most common learner characteristics found in literature [21, 13, 10] which can be used in steering an adaptive learning system. For the preservation of the learner's knowledge we used what is classified as an overlay model in [13]. Until now, the IEEE Learning Object Model standard LOM is considered to be the standard for many repositories storing thousands of learning objects with metadata. There have been attempts to transform the LOM metadata model into an RDF version (e.g., [18]). However, the model provided by LOM was not sufficient. Hence, we adopted part of the LOM model (by relying on previous LOM RDF efforts) and extended it with our own needs.

Our realizations in this project largely replace the functionality of the restrictive SCORM standard [1]. SCORM, an abbreviation for Sharable Content Object Reference Model, is a collection of specifications imposing a format for bundling Web-based exercises into courses, thereby imposing LOM for the metadata, as well as a data model for communicating learning scores between server and client. The standard was updated in 2004, now supporting a limited set of instructions for adaptive behavior. In practise however, the imposed syntax for adaptivity had low expressivity but remaining very complicated [14]. Although in the past SCORM had an important impact on the sharing of bundled learning courses on the web and although many tried to improve the SCORM standard [16, 22, 29], we think its starting point has become outdated. After all, we believe grouping learning objects in a container format conflicts with the principle of the Semantic Web of data in which objects are scattered over the web. Additionally, its extensibility pointed out to be low [14, 6] and the data model for exchanging learning results is limited to the exchange of a single score, thereby not fulfilling our needs of more advanced reporting of a learner's interactions with the exercises. Our formalized representation model for recording scores and interactions with exercises makes it possible to develop true interoperable exercises that are able to report learning results in a universal way. Until now, the importance for adaptive learning systems having an extendible although universally understandable learning result reporting system was largely ignored.

Gang *et al.* proposed a framework for mobile learning in [9] that approaches the challenges similarly as we did here. More specifically, a media delivery system was developed, as well as an adaptive module for learning item selection. However, they relied on MPEG-21 technology while we use the NinSuna platform, which is based on MPEG-21 principles but proven to be more efficient and generic [26]. Further, learning item selection is not based on educational properties such as skills or experience, but solely on environmental properties.

6 Conclusions and Future Work

In order to exploit the possibilities of Web-based e-learning environments, we proposed an e-learning architecture that is able to provide rich, personalized e-learning experiences to a wide range of devices. We discussed the various data models used within the e-learning framework. Moreover, we provided details of the reasoner, a crucial component allowing to select learning items based on the usage environment and the learner profile.

Future work consists of exploiting the possibilities of the Semantic Web even more by linking learning items to the Linked Open Data cloud. Further, data models could be optimized and linked to upcoming efforts (e.g., how to represent the life cycle of a learning item as provenance information on the Web). Also, more detailed domain models should be investigated. Regarding the reasoner, future work consists of taking into account more information obtained from the logging framework, as well as investigating how error-specific feedback could be generated (e.g., link frequently occurring errors to answers).

Acknowledgments

The research activities as described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT, 50% co-funded by industrial partners), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

References

1. Scorm 2004 4th edition version 1.1 overview, http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Overview.aspx
2. Standard for learning object metadata, http://standards.ieee.org/findstds/standard/1484.12.1-2002.html
3. Grapple, a generic responsive adaptive personalized learning environment. http://www.grapple-project.org (Jun 2010)
4. European language levels - self assessment grid (2011), http://europass.cedefop.europa.eu/LanguageSelfAssessmentGrid/en
5. Baker, F.B., Kim, S.H. (eds.): Item Response Theory: Parameter Estimation Techniques, Second Edition (Statistics: A Series of Textbooks and Monographs). CRC Press, 2 edn. (July 2004)
6. Bohl, O., Scheuhase, J., Sengler, R., Winand, U.: The sharable content object reference model (scorm) - a critical review. In: Computers in Education, 2002. Proceedings. International Conference on. pp. 950 – 951 vol.2 (2002)
7. Chen, C.M., Lee, H.M., Chen, Y.H.: Personalized e-learning system using item response theory. Computers & Education 44(3), 237 – 255 (2005)
8. Chi, Y.L.: Ontology-based curriculum content sequencing system with semantic rules. Expert Syst. Appl. 36, 7838–7847 (May 2009)
9. Gang, Z., Zongkai, Y.: Learning Resource Adaptation and Delivery Framework for Mobile Learning. In: Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference (October 2005)
10. Jia, B., Zhong, S., Wang, W., Yang, B.: The construction and evolution of learner model in adaptive learning system. Computer Technology and Development, International Conference on 1, 148–152 (2009)
11. Lee, W., Bürger, T., Sasaki, F., Malaisé, V., Stegmaier, F., Söderberg, J. (eds.): Ontology for Media Resource 1.0. W3C Working Draft, World Wide Web Consortium (June 2010)
12. Leutner, D.: Instructional design principles for adaptivity in open learning environments. Curriculum, Plans, and Processes in Instructional Design: International Perspectives pp. 289–307 (2004)
13. Loc Nguyen, P.D.: Learner model in adaptive learning. World Academy of Science, Engineering and Technology 45 (2008)
14. Mackenzie, G.: Scorm 2004 primer, a (mostly) painless introduction to scorm. Tech. rep. (2004), http://www.pro-ductivity.com/Compliance21CFR/CTMW/scormintro.pdf
15. Mannens, E., Van Deursen, D., Troncy, R., Pfeiffer, S., Parker, C., Lafon, Y., Jansen, J., Hausenblas, M., Van de Walle, R.: A URI-Based Approach for Addressing Fragments of Media Resources on the Web. To appear in Multimedia Tools and Applications – Special Issue on Multimedia Data Semantics

16. Melia, M., Barrett, R., Pahl, C.: A model-based approach to scorm sequencing. In: In Proceeding of the Sixth Annual Irish Educational Technology Users Conference (EdTech06) - Research Track. ILTA (2006)
17. Min, W.X., Wei, C., Lei, C.: Research of ontology-based adaptive learning system. In: Proc. Int. Symp. Computational Intelligence and Design ISCID '08. vol. 2, pp. 366–370 (2008)
18. Nilsson, M., Palmer, M., Brase, J.: The LOM RDF binding - principles and implementation. In: Proceedings of 3rd Annual Ariadne Conference (2003)
19. Perez, M.M., Cornillie, F., Senecaut, M.P., De Wannemacker, S., Desmet, P.: Mobile vocabulary learning: activities allowing for interaction through input enhancement. In: De Wannemacker, S., Clarebout, G., De Causmaecker, P. (eds.) Interdisciplinary Approaches to Adaptive Learning: A Look at the Neighbours,. pp. 15–27. Springer-Verlag (2011)
20. Rani, S., Ashok, M., Palanivel, K.: Adaptive content for personalized e-learning using web service and semantic web. pp. 1 –4 (jul 2009)
21. Roll, I., Baker, R.S., Aleven, V., Koedinger, K.R.: A metacognitive act-r model of students' learning strategies in intelligent tutoring systems. In: Intelligent Tutoring Systems. pp. 854–856 (2004)
22. Silva, L., Mustaro, P.N., Stringhini, D., Silveira, I.F.: Using conceptual lattices to represent fine granular learning objects through scorm meta-objects. The Electronic Journal of e-Learning 4, 141–148 (2006)
23. Soylu, A., Vandewaetere, M., Wauters, K., Jacques, I., De Causmaecker, P., Desmet, P., Clarebout, G., Van Den Noortgate, W.: Ontology-driven Adaptive and Pervasive Learning Environments - APLEs: An Interdisciplinary Approach. In: De Wannemacker, S., Clarebout, G., De Causmaecker, P. (eds.) Interdisciplinary Approaches to Adaptive Learning: A Look at the Neighbours,. pp. 99–115. Springer-Verlag (2011)
24. Troncy, R., Mannens, E., Pfeiffer, S., Van Deursen, D. (eds.): Media Fragments URI 1.0. W3C Working Draft, World Wide Web Consortium (January 2011)
25. Van Deursen, D., Van Lancker, W., De Neve, W., Paridaens, T., Mannens, E., Van de Walle, R.: NinSuna: a Fully Integrated Platform for Format-independent Multimedia Content Adaptation and Delivery based on Semantic Web Technologies. Multimedia Tools and Applications – Special Issue on Data Semantics for Multimedia Systems 46(2-3), 371–398 (January 2010)
26. Van Deursen, D., Van Lancker, W., De Bruyne, S., De Neve, W., Mannens, E., Van de Walle, R.: Format-independent and Metadata-driven Media Resource Adaptation using Semantic Web Technologies. Multimedia Systems 16(2), 85–104 (2010)
27. Wang, Y.K.: Context awareness and adaptation in mobile learning. In: Wireless and Mobile Technologies in Education, 2004. Proceedings. The 2nd IEEE International Workshop on. pp. 154 – 158 (2004)
28. Wauters, K., van den Noortgate, W., Desmet, P.: The use of irt for adaptive item selection in item-based learning environments. In: Proceeding of the 2009 conference on Artificial Intelligence in Education. pp. 785–786. IOS Press, Amsterdam, The Netherlands, The Netherlands (2009)
29. Zhu, X.H., Yan, R.Q., Wu, L.P., Zhang, L.F.: An interoperable learning sequencing model based on the flow control object. Computer Science and Software Engineering, International Conference on 5, 1217–1220 (2008)