

RESEARCH

Open Access

Competency-based personalization for massive online learning

Gilbert Paquette^{1*}, Olga Mariño², Delia Rogozan¹ and Michel Léonard¹

* Correspondence: gilbert.

paquette@licef.ca

¹LICEF Research Center,
Télé-université, 5800 St-Denis Street,
Office 1105, Montreal, Quebec H2S
3L5, Canada

Full list of author information is
available at the end of the article

Abstract

This paper investigates the problem of personalization in massive open online courses (MOOC) based on a target competency profile and a learning scenario model built for the course. To use such a profile for adaptive learning and resource recommendation, we need to be able to compare competencies to help match the competencies of learners with those involved in other learning scenario components (actors, activities, resources). We present a method for computing relations between competencies based on a structured competency model. We use this method to define recommendation agents added to a MOOC learning scenario. This approach for competency comparison has been implemented within an experimental platform called TELOS. We propose to integrate these functionalities to a MOOC platform such as Open-edX. We present a personalization process and we discuss the tools needed to implement the process.

Keywords: Adaptivity; Semantic referencing; User modeling; Smart learning environments; MOOC scenarios; Recommendation; Personalization

Introduction - the semantic adaptive web

The advent of Massive Open Online Course (MOOC) (Hollands et al. 2014; Daniel et al. 2012) raises the issue of personalization even more acutely than before. The same course can be followed by thousands of learners in various parts of the world; all with different background, knowledge and culture, which makes it difficult, if not impossible, to provide an efficient one-size-fits-all learning environment.

The large number and the diversity of learners preclude providing human tutors, as in distance learning university online courses. One solution for personalization is to add “smartness” to MOOC learning environments by dynamically updating learner profiles, recommend adapted resources and activities and adapt the initial scenario to learner profiles.

Commercially mature recommender systems have been introduced during recent years in popular e-commerce web sites such as Amazon or eBay with similar personalization intents. Yet, according to Adomavicus and Tuzhilin (2005), new developments must “include, among others, the improved modeling of users and items, and incorporation of the contextual information into the recommendation process”. The development of the “Web of data” (Heath and Bizer 2011; Allemang and Hendler 2011) also leads to an “Adaptive Semantic Web” (Dolog et al. 2004b; Jannack et al.

2011) where knowledge about users and context of use enables the personalization of Web resources, including learning scenarios.

The present contribution addresses some of these major issues. We propose to provide a context for recommendation and personalization using a multi-actor learning scenario (or workflow) model. This model provides a structure of the activities executed by actors using various kinds of input resources, producing outcomes and interacting with other actors. It is implemented in the TELOS system (Paquette 2010) where an ontology-based competency model serves to annotate actors, activities and resources, providing a “user and item” model for recommendation or scenario adaptation, according to the learners’ competency profiles.

In section “Competency referencing of learning scenario components”, we present the competency model used for recommendation and adaptation and a simple scenario example which is used to illustrate the main concepts involved in our proposal. Unlike other approaches for an ontology-based recommendation, such as OWL-OLM (Denaux et al. 2005) or Personal Reader (Dolog et al. 2004a), this competency model extends beyond simple taxonomies of concepts. It uses an OWL ontology extended with mastery levels, e.g. generic skills and performance levels applied to knowledge in the ontology.

Based on this model, domain-specific competencies are used to annotate actors, activities and resources in the learning scenario. Having annotated all these elements by a common referential is the first step towards a personalized learning environment. The next challenge is to provide a sound method for comparing competency sets and then to decide on particular actions to take to personalize the learning experience, based on the outcome of this comparison. In section “Competency comparison” we address the problem of competency comparison, providing some heuristics to help match a user’s competency model with the competencies possessed by other actors or involved in activities and resources in the scenario.

In section “Recommendations based on competency comparison”, we present the definition of rule-based advisory agents, where the competency relations defined in section “Competency comparison” are used into the rules’ conditions. The corresponding rules’ actions help personalize the environment by showing or hiding resources, reorienting activity paths, proposing additional elements or pointing toward relevant peers. As a proof of concept, we present a scenario implemented within the TELOS ontology-driven system (Paquette and Magnan 2008).

Section “Personalization of massive open online courses” describes the use of our model and system for personalizing MOOCs. Two forms of personalization are exemplified. The first one uses competency relations to automatically cluster MOOC learners into subgroups that are more manageable for collaboration and for which we can propose adapted scenario versions. The MOOC designer can customize the personalization rules to decide whether to have competency homogenous groups or on the contrary groups with skilled student helping less advanced ones. Secondly, we integrate advisory agents as in the example of sections “Competency referencing of learning scenario components” and “Recommendations based on competency comparison” to recommend adapted activities and resources to individual learners, whatever their subgroup.

In section “Overview of the process and required tools for MOOC personalization”, we conclude by presenting the overall scenario adaptation process from the viewpoint

of the design team. We discuss possible extension of the process and the main tools that will be needed to integrate the whole process in MOOC delivery platforms.

Competency referencing of learning scenario components

We present in this section a learning scenario model that will provide the operational context for recommendation and adaptation. This model is implemented using our graphical scenario editor, G-MOT and our compatible scenario manager and execution engine, TELOS. Basically, the scenario model aggregates actors, the activities they perform and the resources they use and produce during the flow of activities.

Scenario models for learning contexts

Figure 1 shows a simple scenario model. There are four activities or tasks (ovals), two actors (a professor and a student) and some resources (I/P linked) that are input to the activities or produced by the actor responsible for the activity (identified by a R link). Each activity is decomposed into sub-models (not shown on the figure), which describe it more precisely on one or more levels. This scenario will serve to illustrate the concepts presented in this paper.

In the first activity (Start), the student reads the general assignment for the scenario and the list of target competencies he/she is supposed to acquire. In the second one, using the information in a document called “Planet Properties”, the student builds a table of planet properties that is validated by the professor (in a MOOC, this should be automatically done by a software agent). In the third activity, using a version of this “Validated table”, he compares properties of planets to find out relations between them,

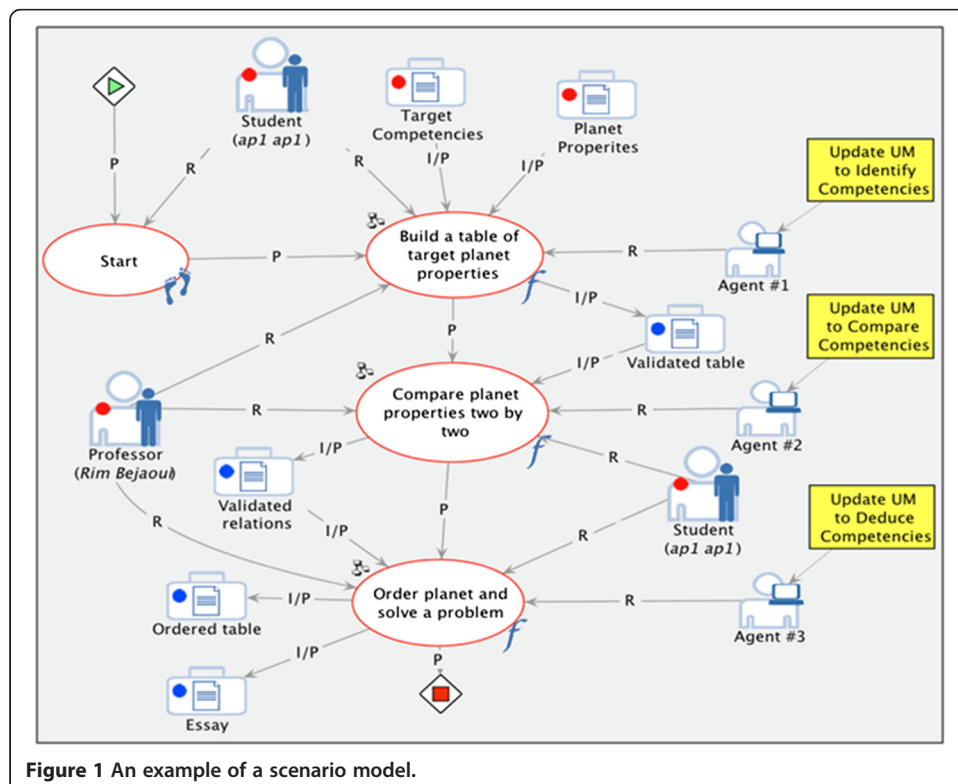


Figure 1 An example of a scenario model.

and writes a text, “Validated relations”, to present his findings. In the last activity he is asked to order planets according to their distance to the Sun and to write his ideas on planets that can sustain life.

On the right side of the figure, three agents (i.e. recommenders or advisor agents) have been inserted to assist each student in executing the corresponding activity, by providing personalized advice. These agents, further explained in section “Recommendations based on competency comparison”, are also used to update each learner’s competency annotations with newly acquired competencies.

Semantic referencing of scenario components

As we have pointed out in (Paquette and Marino 2011), educational modeling languages and standards such as IMS-LD (2003) need to be improved with a structured knowledge and competency representation, in order to add useful semantic annotations to scenario components.

For semantic annotation, two main methods are generally used: annotation with concepts belonging to a taxonomy or annotation with natural language statements defining prerequisites and learning objectives.

We proposed to use a third approach based on competency referencing (Paquette 2007), where a competency can be understood as a skill applied to a knowledge, with a certain level of performance; the knowledge element in this definition being defined in a domain ontology.

A domain ontology needed for semantic annotations can be downloaded from ontology repositories available on the Web for various subjects. Only if the teacher/designer cannot find a suited one must he build it. Ontologies are “shared visualizations” of a knowledge domain, thus largely independent from a particular teacher’s view. What are specific are the skill and performance levels added to ontology components to define target competencies.

Unlike other approaches based on ontologies, such as OWL-OLM (Denaux et al. 2005) or Personal Reader (Dolog et al. 2004a), our model generalizes taxonomy-based annotations with OWL-DL ontologies annotations and adds mastery levels (generic skills and performance levels) to ontology references.

Furthermore, to state only that a person has to “know” a concept is an ambiguous statement. It does not say what exactly the person is able to do with the concept. For example, stating that someone “knows a certain device” may mean competencies ranging from “being able to describe its structure” to “being able to recognize its malfunction” to “being able to repair it”. Also, it is very different if a diagnosis is to be made in a familiar or novel situation, or with or without help; these are examples of performance indicators or criteria adding precision to the statement of a generic skill.

We thus define each competency as a triple (K, S, P) where K is a knowledge element, a class, a property or an individual from a domain ontology, S is a generic skill (a verb) from a taxonomy of skills, and P is the result of combining performance criteria values. Domain ontologies follow the W3C OWL-DL standard. The taxonomy of skills is simplified to a 10-level scale (0-PayAttention, 1-Memorize, 2-Explicitate, 3-Transpose, 4-Apply, 5-Analyze, 6-Repair, 7-Synthetize, 8-Evaluate, 9-Self-Control). The performance part is a combination of performance criteria values providing four

performance levels (0.2-aware, 0.4-familiar, 0.6-productive, 0.8-expert), added to the skill level.

We have developed a *Competency editor* inside the TELOS tool set, to create and manipulate this kind of competency models. To connect competencies from a domain specific competency model with a learning scenario relating to this competency model, we developed another TELOS tool, called the *Competency referencer*. Using this tool, one can specify learners' actual competencies, activities' and resources' entry and target competencies.

For example, using a domain ontology for solar system planets such as the one on Figure 2 and a competency model based on this ontology, competencies can be associated to resources from the scenario (as shown on Figure 3). The entry and target competencies describing such a resource, ("*Planet Properties*"), could be compared with the actual competencies of a user to verify if he has all of them, or some, or none, in his competency model, and offer a recommendation, such as to skip the resource (too easy for this user) or to study a preliminary document (this user doesn't have the competences needed to study it, that is this resource resource entry competencies).

Activities, resources and user competency models

Selected components of a scenario are thus referenced using comparable competencies, based on the same domain ontologies and competency model. Resources and activities in a scenario are referenced by two sets of competencies, one for prerequisite competencies, and the other, for target competencies (i.e. learning objectives).

In (Paquette and Marino 2011), we provided a multi-actor ontology-based assistance model supported by a *user competency model*. This user model is composed of three main parts (Moulet et al. 2008):

- List of the user's actual competencies selected in one or more competency referentials. As mentioned above, each user's competencies (C) is described by its knowledge (K), skills (S) and performance (P) components.

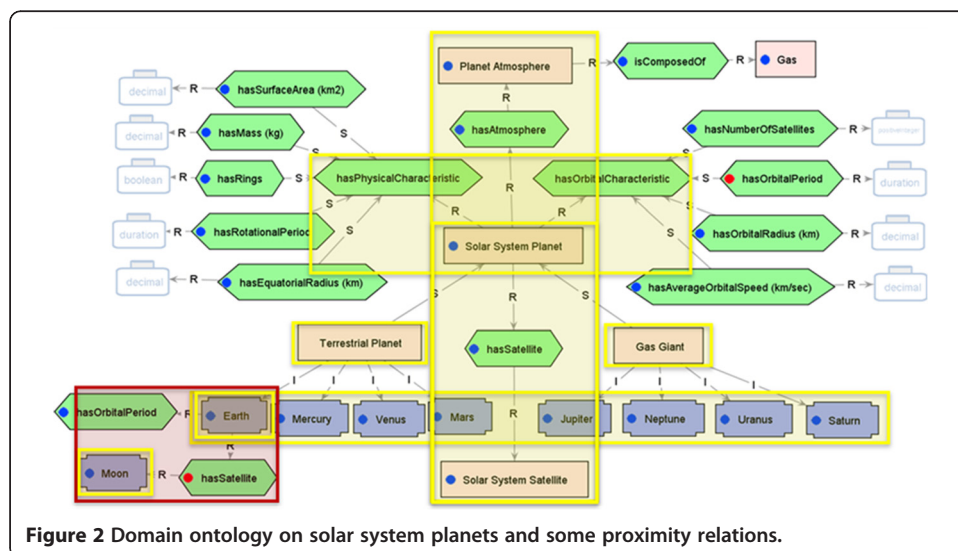


Figure 2 Domain ontology on solar system planets and some proximity relations.

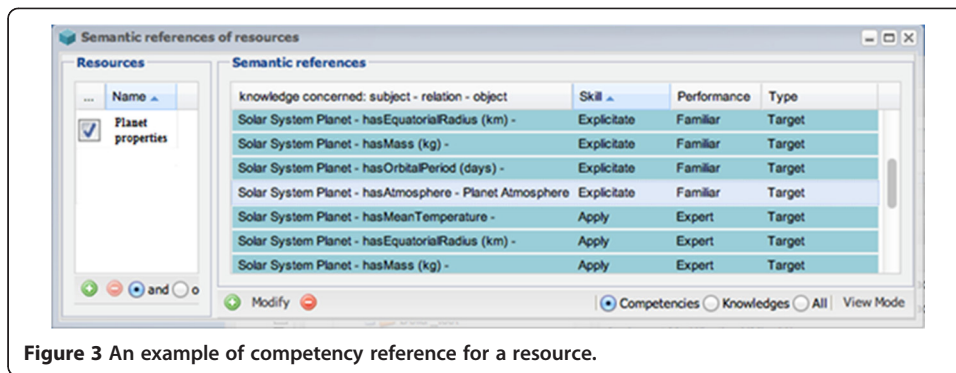


Figure 3 An example of competency reference for a resource.

- Documents (texts, exam results, videos, images, applications, etc.) structured into an e-portfolio that presents evidence for the achievement of related competencies.
- Context in which a competency has been achieved. It includes the date of achievement, the activities that led to it, the link to the evidence in the e-portfolio and the evaluator of this evidence.

Figure 4 shows an example of a user portfolio that contains a list of competencies from two domains, instructional design and solar system planets. On the right side of the figure, evidences for the selected competency are shown as well as possible annotation or comments from an evaluator.

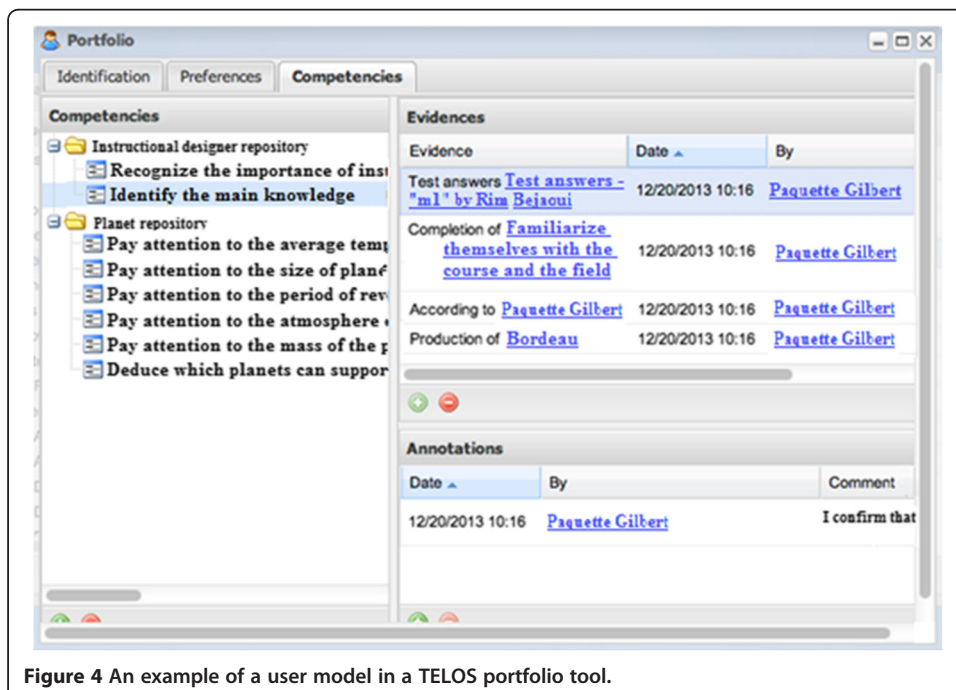


Figure 4 An example of a user model in a TELOS portfolio tool.

Competency comparison

Once actors, activities and resources in a learning scenario have been referenced using a competency profile, we need to be able to compare competencies to find appropriate actors, activities or resources that can support the learners in the scenario. This section will provide such a comparison method that has been implemented within the TELOS system.

Knowledge and competency comparison relations

Consider two competencies $C_1 = (K_1, S_1, P_1)$ and $C_2 = (K_2, S_2, P_2)$. It will be rarely the case that the three parts will coincide, but we can evaluate the semantic proximity or nearness between C_1 and C_2 , based on the respective positions of their knowledge parts in the ontology and the levels associated with the skills and the performance parts.

Our recommendation/adaptation agents will evaluate if a user's actual competency coincides, is very near, near or far from the prerequisite or target competencies of a resource or an activity, as well as compared to the actual competencies of another user. These agents can also evaluate if a competency is stronger or weaker than another one according to the levels of its skill and performance parts. Or it can determine if the competency is more specific or more general according to the positions in the ontology of the corresponding knowledge components.

Thus, to take advantage of the competency model, we need to establish a formal framework for the evaluation of the proximity, strength or generality of competencies. In the next section we define the semantic proximity between knowledge parts of a competency. In section "Semantic relationships between Competencies" we extend the comparison method to competencies by considering skills and performance.

Semantic proximity of the knowledge components

In this section, we focus only on the knowledge part of the competencies to be compared. Maidel et al. (2008), propose an approach in which taxonomies are exploited. Five different cases of matches between a concept A in the resource (or item) profile and a concept B in the user profile are considered. Various matching scores are given when a concept A in the item profile, a) is the same, b) is a parent, c) is a child, d) is a grandparent or e) is a grandchild of a concept in the user profile. Then, a similarity function is used to combine these scores to recommend resources to a user according to his acquired concepts. Maidel et al. (2008) state that if the use of a taxonomy is not considered, the recommendation quality significantly drops.

We agree with this statement but we believe that extending the comparison to ontology components instead of only specialization relations between classes would provide a stronger basis for competency comparison. Note for example that in the solar system domain, property comparisons are more important than class relationships.

We thus propose to define the semantic proximity between knowledge elements, based on their situation in the domain ontology. *Knowledge references* are components from OWL-DL ontologies that describe the knowledge in a resource. A few examples of these knowledge references are shown on Figure 2 that presents part of an ontology for solar system planets^a. Knowledge references can take six different forms: *SolarSystemPlanet* is a class reference (C); *Neptune* is an instance reference (I); *SolarSystemPlanet/hasAtmosphere/Atmosphere* is an object property reference with its domain and range classes (D-oP-R);

Earth/hasSatellite/Moon is an object property instance reference (I-oP-I); *SolarSystem-Planet/hasOrbitalPeriod* is a data property reference with its domain class (D-dP); *Earth/hasNumberOfSatellites* is a data property instance reference (I-dP).

We have investigated systematically these six forms of OWL-DL references to decide on the nearness of two references K_1 and K_2 . For example, a concept (form C) is near its sub classes, super classes, and instances. It is also near an object or data property (forms D-oP-R and D-oP) that has a domain or range identical or equivalent to this concept. A property reference, with its domain and range (form D-oP-R) is near a sub-property or super-property with the same domain and range. It is also near to a sub-property or a super-property with a subclass or superclass of its domain and range.

Another comparison criteria is concerned with a reference K_1 being more general or more specific than another one K_2 . For example, K_1 is more general than K_2 if K_1 is a superclass of K_2 , or has K_2 as an instance, or appears as domain or range of a data or object property reference K_2 , or contains an instance in the domain or range of a data or object property instance reference K_2 .

Semantic relationships between competencies

We now extend the comparison to competencies, adding the skill (S) and performance (P) components of the competency model. Figure 5 presents a few comparison cases between two competencies $C_1 = (K_1, S_1, P_1)$ and $C_2 = (K_2, S_2, P_2)$ in the case where K_1 is near K_2 . Other cases are not considered, i.e. comparison fails.

To illustrate the heuristics, the (S, P) couples are represented on a 2-dimensional scale in Figure 5. Skills are ordered from 0 to 9 and grouped into four classes as follows: $\{\{0,1\}, \{2,3,4\}, \{5,6,7\}, \{8,9\}\}$. Performance indicators are grouped into four decimal levels.

For example, a competency C_1 with an analyze skill at an expert level is labeled 5.8 ($S_1 + P_1$). A competency C_2 at a level 7.2 or 6.4 will be considered near and stronger than C_1 because the synthesize skill or the repair skill are in the same class than the analyze skill, but one or two levels higher in the skill’s hierarchy. On the other hand,

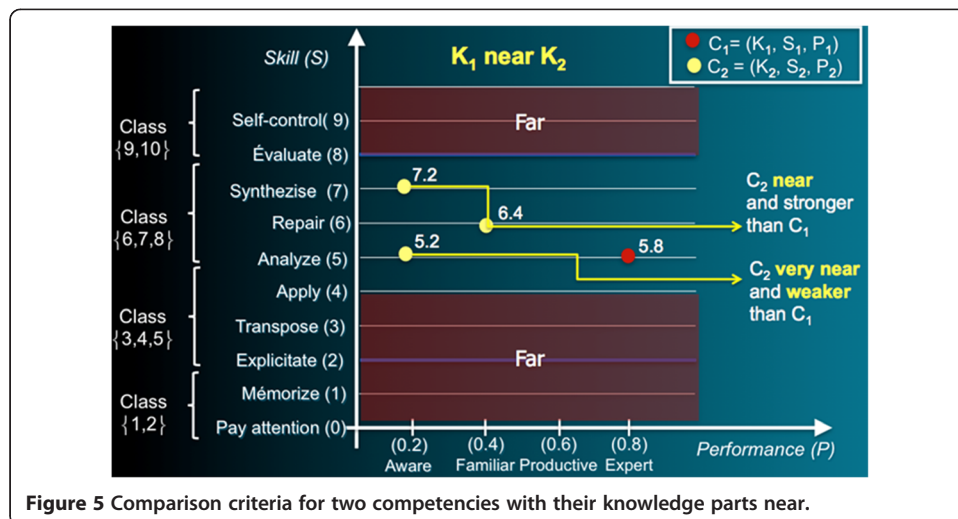


Figure 5 Comparison criteria for two competencies with their knowledge parts near.

a competency C2 at a level 5.2 will be considered very near and weaker than C1 because it has the same skill's level but with a lower performance level. Other possible competencies in the “far zone” will be considered too far to be comparable.

Also, depending on the relationship between K1 and K2, C2 will be defined as equivalent, more general or more specific than C1. These relations between competencies can also be combined to define more complex relationships. For instance, it is possible for a competency reference to be at the same time identical or very near/near, stronger/weaker and more general/more specific than another one.

We have developed an inference system that deduces these relations from a competency referential edited by a competency editor as presented above.

Recommendations based on competency comparison

In this section, we show how we use the competency comparison relations into the rule conditions of the recommendation/adaptation software agents.

Competency-based conditions and rules

Recommendation agents are added to a scenario, linked to some of the activities or functions (grouping of activities) called *insertion points*, as shown in Figure 1. The designer defines these agents by creating a set of rules. In each rule, one and only one of the actors linked (using R links) to the activity at the insertion point is chosen as the receiver of the recommendation. If a triggering event occurs at run time such as “activity completed”, “resource opened”, etc., each applicable rule condition is evaluated and its actions are triggered or not, depending on the evaluation of the condition.

A competency-based condition takes the form of a triple:

- *Quantification* takes two values: HasOne or HasAll, which are abbreviations for “the receiving user has one (or has all) of its competencies in some *relation* with an object competency list”.
- *Relation* is one of the comparison relations between semantic references presented above: Identical, Near, VeryNear, MoreGeneric, MoreSpecific, Stronger Weaker or any consistent combination of these relations.
- *ObjectCompetencyList* is the list of prerequisite or target competencies of the activity at the insertion point or of a resource input or output of this activity.

Let's take the example of a condition like:

(user) HasAll (its competencies)/Near and MoreSpecific/(than) Target competencies for Essay

When this condition is evaluated, competencies in the user's model, for the specified user at the insertion point, are retrieved, together with the list of target competencies for the resource “Essay”. The evaluation of the relation “Near and MoreSpecific” provides a true or false value according to the method exposed in section “Semantic relationships between Competencies”.

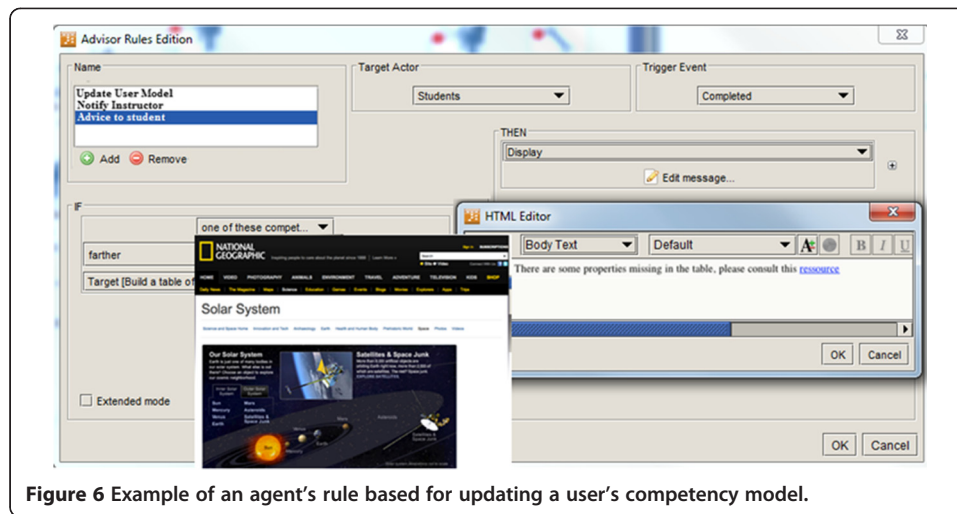


Figure 6 Example of an agent's rule based for updating a user's competency model.

Recommendation actions

The action part of an agent's rule can perform a number of recommendations or adaptations in a scenario: give advice to an actor, notify another actor than the one targeted by the rule, recommend various learning resources, update the user's model, and propose to jump to another activity or to another learning scenario.

All these possibilities have been implemented in TELOS. On Figure 1, we have presented a simple scenario with three recommendation agents. For example, Recommender agent #1 on Figure 1 will verify if the learner has succeeded the second activity in the scenario ("Build a table..."). It has 3 rules, shown on the screen-shot of Figure 6: update model, notify instructor or advice to learner.

The rule "Update User Model" transfers the list of target competencies associated to the activity to the user's model, if he has succeeded the activity, and built a validated table of planet properties. If he has failed, the second rule will send a notification to the professor to interact with the user. Finally, a third rule (selected on Figure 6) provides an advice to a user that has partly succeeded the activity, recommending the resource shown on the figure.

Multi-agent recommendation system

The recommendation system is epiphyte to the learning system, implying that it can evolve and change without modifying the learning scenario. One can even have various different recommendation systems for a same learning scenario, possibly for teachers of the same course with different pedagogies. The system is based on production rules of the form < actor, event, condition, action >. A particular action "delegate" allows the propagation of an local event and its activation context to other scenario parts, for instance to adapt the final exam content, based on a particular activity performance.

Personalization of massive open online courses

We will now address a typical MOOC scenario to illustrate a method for personalizing a massive open online course. We aim to identify possible bottlenecks and critical tasks where new or adapted tools are needed for the TELOS system or for MOOC platforms

like Open-edX (2014). The goal here is to add to an initial MOOC scenario a number of software advisors, as explained in the previous section, and a module that will help cluster learners into subgroups where learners have similar competency profiles to provide them with an adapted scenario.

The MOOC global scenario

Figure 7 present a MOOC scenario for a course on interviewing techniques to which a preliminary module has been added for personalization. Each of the five modules (represented by single white ovals) spans on 1–2 weeks.

- Module 0 proposes preliminary activities evaluated using various methods to subdivide the group of learners into subgroups having similar competencies; e.g. three subgroups termed “novice”, “intermediate” or “advanced”. There could be more subgroups but three seems sufficient for our illustration purpose.
- When the subgroups are created, each learner will afterwards (P link) do an activity (represented by multiple ovals) where he/she will choose a plan for the rest of the course adapted to his/her subgroups’ competency profile. This plan is produced by an advisory agent that will propose an ordering of modules A, B and C, according to the results of module 0. For the advanced subgroup for example, module B can be skipped; for the intermediate subgroup, module B is proposed after module C as enrichment; for the novice group, module A, B and C are done in that sequential order.
- For all three subgroups, module D follows the other modules completion.
- For each of module A, B, C and D, the activities and the resources proposed to learners will differ from one subgroup to the other, according to the recommendations of the corresponding advisory agents.

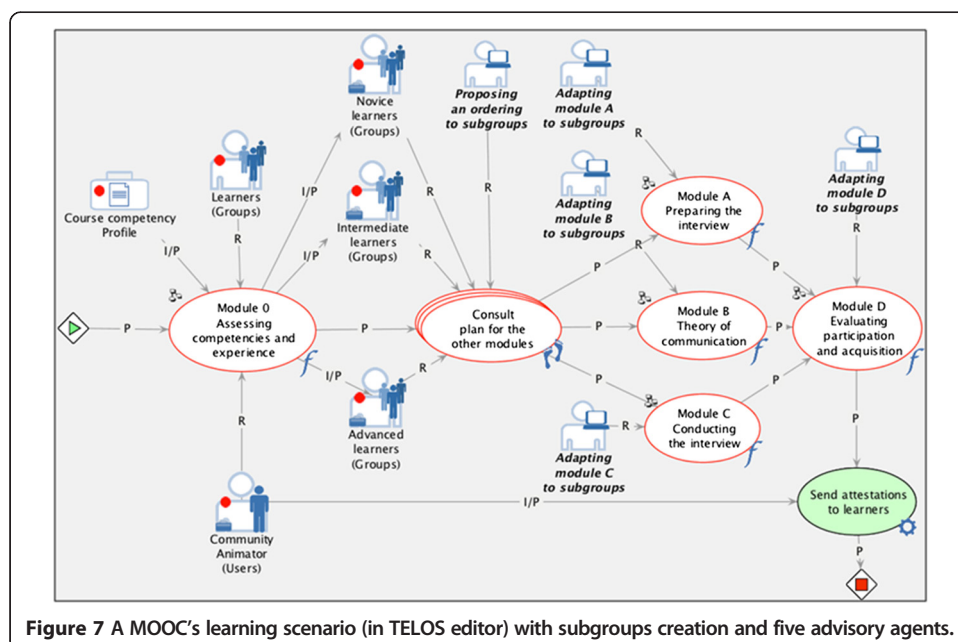


Figure 7 A MOOC’s learning scenario (in TELOS editor) with subgroups creation and five advisory agents.

- Another actor, the “Community Animator”, participates in module 0 and also triggers the operation that sends automatic attestations to learners, after module D is completed.

Creating the subgroups

We will now expand on Figure 8 the sub-model for module 0 where the subgroups are created to help personalize the learning scenario. This module starts when the community animator sends a notification to the learners to start the module’s first activity. Before that, the community animator will have produced a self-evaluation questionnaire to be used by learners to assess their own competencies. For this, he triggers the *Q&A automatic production* operation of Figure 8. This operation takes as input a *Course competency profile* such as the one shown on Figure 9 and produces a *Competency self-evaluation questionnaire*. This questionnaire presents each competency in the profile to each learner and asks if he/she considers him/herself at one of these four levels: aware, familiar, productive, or advanced. These levels are defined according to the same competency performance criteria shown on Figure 9 and the definition of the four levels are provided to the learners to help them select the right level that corresponds to their situation.

This questionnaire will be used in the third activity of module 0. In the first activity, each learner consults video and/or texts on interviewing techniques. Afterwards, they

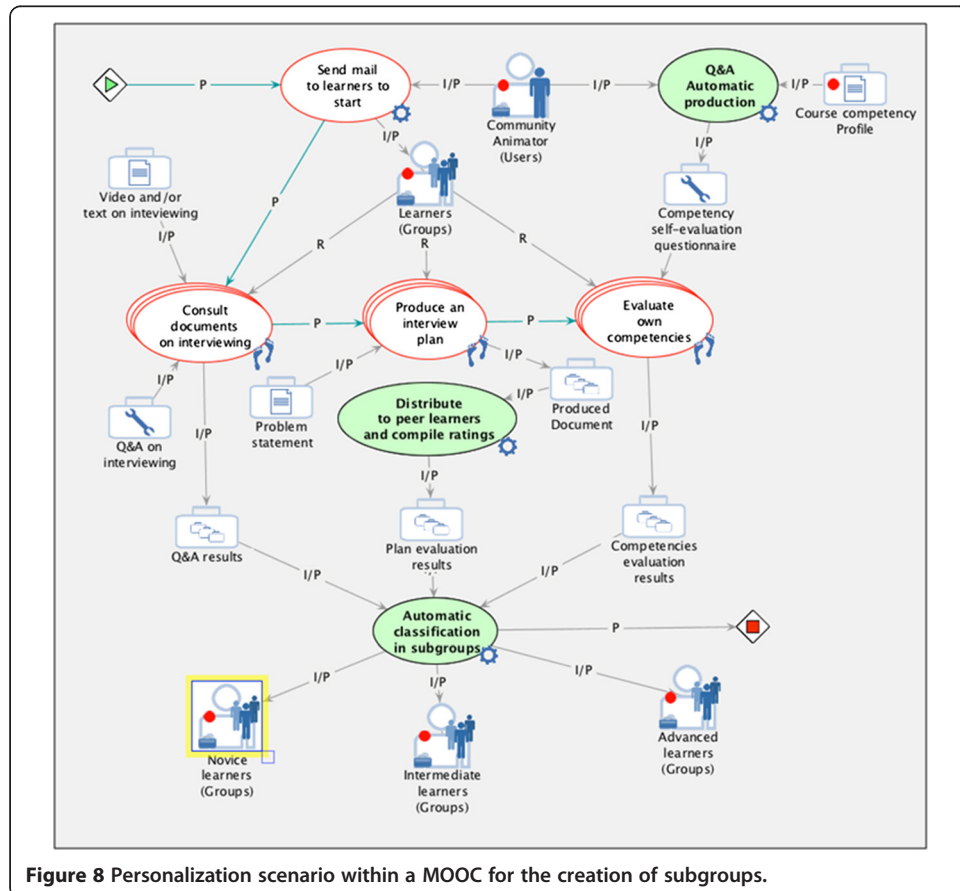


Figure 8 Personalization scenario within a MOOC for the creation of subgroups.

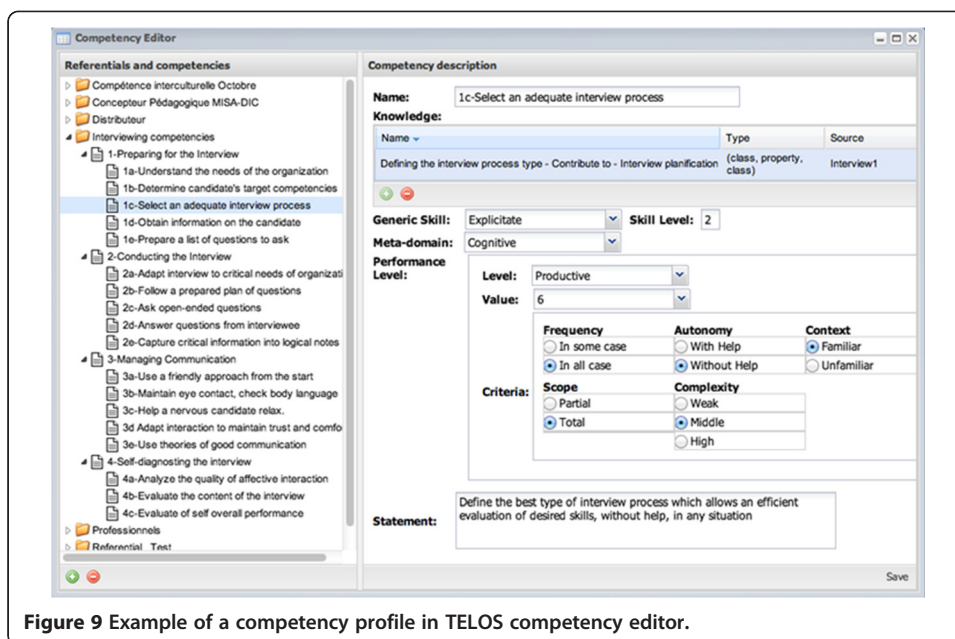


Figure 9 Example of a competency profile in TELOS competency editor.

pass a *Q&A test on interviewing techniques*, and the results are compiled automatically for each learner. MOOC platforms usually provide such operations to assess learners.

The second activity engages learners to write an interview plan. The input is a *Problem statement* where an organization aims to recruit new professionals to achieve strategic functions within the organization. The *Produced document* will be automatically and randomly distributed to a number of peer learners who will rate the documents assigned to them using a questionnaire related to the competency profile for the course. The *Plan evaluation results* of the second activity will be compiled by the operation *Distribute to peer learners and compile ratings* shown of Figure 8.

In the third activity, each learner will assess his/her own competencies using the *Competency self-evaluation questionnaire* prepared by the course designer as mentioned before, therefore producing the *Competencies evaluation results* shown on Figure 8.

Finally, an operation for “Automatic classification in subgroups” will combine the results data of the three activities, according to some weighting policy established by the course designer, producing the list of learners for each subgroup and sending a notification to each learner to inform them on their assigned subgroup.

Then the execution of module 0 stops and the flow of control in the scenario moves back to the global scenario and to the other activities and modules of the MOOC.

Competency referencing of actors, modules and activities, and resources

Competency referencing plays a major role in the proposed personalization process. To start the process, a specific competency profile or model must be built according to the course content.

Such a profile, named here *Interviewing competencies* is built using the TELOS competency editor shown on Figure 9. Other competency profiles have been built for other course subject. Each profile follows the model presented in sections “Semantic referencing of scenario components” and “Activities, resources and user competency models”. First the

knowledge part is declared, then a generic skill is selected with its type (or meta-domain) and finally the performance criteria are set, combined automatically to produce the performance level for each competency.

The Interviewing competencies' profile is shown on the left part of Figure 9. One of the target competencies for the course labeled "1c-Select an adequate interview process" is selected for edition and displayed on the right side of the figure. Its knowledge part is a property of an interview process with its domain and co-domain classes; its generic skill part is *level 2-Explicitate* and is typed "*cognitive*"; its performance part is at the level *6-productive* resulting from the combination of the five criteria below. These competency parts are combined in a precise natural language statement: *Define the best type of interview process that allows an efficient evaluation of desired skills, without help, in any situation.*

This competency profile has four sections corresponding to the various modules of the course. These modules are referenced by the competencies in corresponding sections of the profile; for example, module A is referenced by all the competencies in section "Preparing for the interview". Inside module A, the activities and resources are referenced by only part of the competencies for the module. It is up to the designer to ensure that all the competencies of the module are well covered and well distributed amongst the activities and the resources that compose the module.

The learner's competencies are set initially at the end of the preliminary module 0. At the end of this module, initial competencies have been attributed to the different learners during the classification operation that has produced the subgroups of learners. These initial competencies of individual learners will evolve throughout the course scenario.

Consider an activity in module A that consists in selecting an adequate interview process. After evaluation of this activity, depending on the quality of the outcome, a competency similar to 1c, possibly with a lower performance level will be added to the learner's model in a similar way as explained in section "Multi-agent recommendation system". Also, an improvement activity will be proposed by an advisory agent to this learner by providing additional resources to consult. Hopefully, at the end of the course, the set of actual competencies of this learner will reach the targeted levels in the competency profile for the course.

Overview of the process and required tools for MOOC personalization

In this concluding section we provide an overview of an adaptation/personalization process that is integrated in the overall instructional engineering process. We also discuss some strategic tools that need to be developed or adapted to make the adaptation/personalization process efficient for MOOCs.

The adaptation/personalization process

We now focus on the work of the design team. Figure 10 presents an abstracted process from the example of the previous section. It starts with the "Original scenario" and leads to the production of an "Adapted scenario".

In a first activity, the design team will prepare a competency profile using a *competency editor*. The TELOS competency editor presented on Figure 9 is one solution that provides a natural language statement of the competencies, but more important it provides their



Figure 10 The general scenario adaptation process.

underlying structure (K, S, P). Simpler solutions exist such as an outline/tree editor that is available in any text editor, but it would have to be extended to provide the structural parts of a competency on which competency comparison relations can be computed. One could also use a simple taxonomy of concepts instead of a complete OWL ontology as we have done in the examples here.

In the second activity we need to reference the MOOC modules with target competencies from the profile defined in the first activity. For this we need a *semantic referencing tool* to assign target competency at least to each module of the course. The same tool would also enable designers to assign entry competencies to modules in order to notify learners who would not have the capacity to start a module and to recommend preparatory activities in this case. Inside the module, the design team would also annotate with competencies critical activities and resources for which recommendations seem necessary.

The next two functions on Figure 10 represent the two methods of adaptation employed in the example of section “Personalization of massive open online courses”. They are detailed in sub-graphs that we will discuss in the following paragraphs.

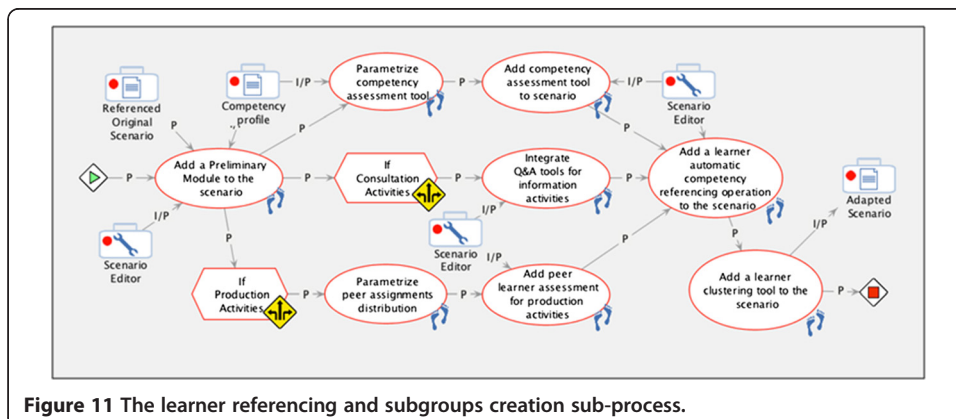


Figure 11 The learner referencing and subgroups creation sub-process.

Adding a preliminary module for creating subgroups

The first adaptation method is adding a preliminary module containing one or more ways to evaluate the competencies of each learner. The evaluation results serve to define subgroups of learner with similar, compatible or complementary competencies, as well as provide a basis for identifying the competencies of individual learners. Figure 11 presents this sub-process.

The first design activity is to add a preliminary module to the original scenario and reference the learning activities and resources within that module with competencies as was done previously for the rest of the course. The new module will be created using a *scenario editor*. It can be the same editor used to build the original scenario or another one. For example, the new module could be built with Open-edX studio if the original scenario had been built in this editor. The new module could also be integrated in Open-edX by an external call to another scenario editor obtained by specializing the TELOS editor used in section “Personalization of massive open online courses”.

There are many options for the learning activities that can be inserted in the preliminary module to evaluate learners’ competencies. In the example of section “Personalization of massive open online courses”, we presented three main options that we generalize in the design sub-process of Figure 11.

The first one is compulsory. It consists in automatically producing a questionnaire from the competency profile to enable each learner to rate his own competencies for the course. This competency questionnaire can be very simple, for example a set of multiple-choice question based on the competency statements with choices such as {unaware, familiar, productive, advanced} or any other set of value consistent with the way competencies are defined. In this way, a distance between the actual competencies of learners and the target competencies for a module, activity or resource can be automatically computed for each learner. The same kind of competency comparison can be done to check if a learner has the prerequisite competencies for a module, activity or resource.

One might think it is not sufficient to have learners evaluate their own competencies. Some learners might overestimate or underestimate themselves. Still, there is an incentive for a learner to be as precise as possible because providing the best information possible opens the possibility for better assistance to the learner from the system.

But additional ways to assess competencies are offered with the two other options of Figure 11. In the first case, if video or text consultation activities are integrated to the preliminary module scenario, a simple Q&A test can be added, producing evaluation results automatically. In the second case, if there are production or construction activities (essays, problem solutions, graphs,...), the resulting productions can be distributed to peer learners who will rate a limited number of their peers by filling a small assessment questionnaire. In both cases, the Q&A and the peer assessment questionnaires must be related to the competency statements for the course, so that the results can be combined with the self-assessment results.

If two or more evaluation activities are inserted in the MOOC preliminary module, we need to add some automatic operations in the system, one of the features in TELOS. One operation simply combines the evaluation results using a weighting policy before adding the result in the learner’s competency references. Another one is the

integration of a *learner clustering* operation. This important operation combines the various competency evaluation results using a clustering policy specified by the designers. The policy will include at least the relative weights assigned to the various evaluation methods, and a specified number of subgroups to be created. The tool will create automatically the lists of subgroup members and notify the individuals on their membership and their peers' average competencies.

More advanced parameters can be included in the clustering policy. One way would be to integrate in the same group learners with similar cognitive competencies, but with complementary technical or socio-affective competencies, so that collaboration inside a group can be more productive. Another extension is to use historic data from previous instances of the course, using learning analytics methods, to cluster learners according to a clustering policy.

Inserting recommenders/advisory agents in the scenario

Finally, to conclude the adaptation process, the design team will use a scenario editor to insert recommenders or advisory agents for critical modules, activities or resources in the course scenario in a way similar to the previous sections.

Right after the preliminary module, when the subgroups have been created, the design team can insert a *plan recommender agent* to propose to each subgroup an adapted path within the course. For this, the initial scenario must be designed in a modular way so that the modules can be followed in different orders, and also the activities and resources within a module. Also options can be provided between modules and/or activities and resources. This leads to a number of alternative plans that can be proposed to the various subgroups.

Next, the design team will identify critical modules and activities where *resource recommender agents* are inserted. At these corresponding insertion point, additional or alternative resources can be proposed. Also, *notification recommender agents* can be added to propose a change in the study plan. As long as a learner succeeds in some activities, his actual competencies will evolve and he must be notified of the change. Also, sometimes, the evolution of a learner's competencies will create a situation where he must be notified to switch to another subgroup plan.

Conclusion

In this paper we have presented a process for the adaptation of learning scenarios and resources that we propose for semantic personalization of MOOC learning experiences. This paper is a contribution to the general field of Differentiated Instruction. Differentiated instruction is a framework or philosophy for effective teaching that involves providing different students with different avenues to learning. As proposed by (Tomlinson 1999), by considering varied learning needs, teachers can develop personalized instruction so that all children in the classroom can learn effectively.

We address this important preoccupation within the Adaptive Semantic Web framework by pretesting MOOC learners on their actual competencies within a shared competency model to form sub-groups with differentiated learning scenarios. Within these scenarios, we provide, through a production rule system, various learning activities and resources according to the gap between learners' actual competencies and the target competencies for the course.

This approach is based on a competency model extending domain ontologies with skills and with performance criteria, which serves as a common semantic referential for activities, resources and learners in the learning scenario.

Several editing, execution and inference tools have been developed to support the model and the design process. As a proof of concept, we have tested the model and the tools on a limited number of scenarios. Although the definition of the agent's rules might be a demanding task, it is worth noticing that only a few activities per scenario may need to be personalized to provide for a personal learning experience. Moreover, recent work on semantic annotation of learning objects and repositories as well as on e-portfolios standards open the door to automatic extraction of semantic information both of the resources and of the learner.

The context of a MOOC provides the advantage of a large number of learners. Having access to large sets of learner data will help refine the relation for semantic nearness between OWL-DL references. Adding weights to the various cases would probably improve the quality of the evaluations. For example, one could assert that a subclass or superclass is closer to a class than its instances or one of its defining properties, especially if there are many defining attributes for this class. This is an interesting path to be investigated.

Another important future work is the method's feasibility in practice. The model presented here is quite general and thus, it is not that simple to implement. We need to identify important special cases in a MOOC context that are manageable for average designers. To improve the practical use of approach in a MOOC, most of the tasks will have to be at least partly automated to take in account that a human tutor approach is not feasible. In the actual system, relations used in the rule conditions are derived automatically from the ontology and competency structures. Also, the semantic tagging of learner is done automatically through various forms of competency assessment. This is part of the solution but we do not pretend that we have a complete practical solution. Automatic or semi-automatic extraction of semantic tag for resources could be achieved using text analysis techniques for example.

Still, the approach presented here sets the ground for an open and flexible method to personalize MOOCs. This is a major issue for the quality of the learning environments, which depends mainly on the capability to adapt course content seamlessly to learners' prerequisite competencies and goals.

Endnote

^aUnlike other graphic presentation of ontologies, properties are shown as objects (hexagons) between their domain and range classes (rectangles). In this way, the relations between properties can be shown on the same graph. Individual are linked to classes by an "I" link.

Competing interest

The authors declare that they have no competing interests.

Authors' contributions

GP wrote the first version and revised all versions of the manuscript. OM added precisions to the text of the manuscript. DR and ML provided some of the figures. DR transform the manuscript to the required format. All authors read and approved the final manuscript.

Author details

¹LICEF Research Center, Télé-université, 5800 St-Denis Street, Office 1105, Montreal, Quebec H2S 3L5, Canada.

²Department of system engineering and computation, Engineering Faculty, Univesidad de los Andes, Bagota, Colombia.

Received: 29 August 2014 Accepted: 21 January 2015

Published online: 17 February 2015

References

- G Adomavicius, A Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
- D Allemang, J Hendler, *Semantic web for the working ontologist*, 2nd edn. (Morgan-Kaufmann, Waltham, MA 02451, USA, 2011)
- J Daniel, Making Sense of MOOCs: musings in a maze of myth, paradox and possibility. *J. Interact. Media Educ.* (2012), <http://www-jime.open.ac.uk/article/2012-18/html>. Accessed 19 August 2014.
- R Denaux, L Aroyo, V Dimitrova, An approach for ontology-based Elicitation of User models to enable personalization on the semantic web. Paper presented at the 14th international world wide web conference, Chiba, Japan, 10–14 May, 2005.
- P. Dolog, N Henze, W Nejdl, M Sintek, The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies. Paper presented at the 3rd international conference on adaptive hypermedia and adaptive web-based systems, Eindhoven University, Netherlands, 23–26 August 2004a.
- P Dolog, N Henze, W Nejdl, M Sintek, Towards the adaptive Semantic Web. *Lecture Notes in Computer Science* (Springer, 2004b). http://link.springer.com/chapter/10.1007%2F978-3-540-24572-8_4#page-1
- T Heath, C Bizer, *Linked Data – Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology* (Morgan and Claypool, 2011). <http://info.slis.indiana.edu/~dingying/Teaching/S604/LODBook.pdf>.
- F Hollands, D Tirthali, MOOCs: expectations and reality. (Columbia University Teachers' College, 2014), http://cbcse.org/wordpress/wp-content/uploads/2014/05/MOOCs_Expectations_and_Reality.pdf.
- IMS_LD: Learning Design Specification (2003), <http://www.imsglobal.org/learningdesign/index.cfm>. Accessed 19 August 2014.
- D Jannack, M Zanker, A Felfering, G Friedrich, *Recommender systems, an introduction* (Cambridge University Press, Cambridge, Mass. USA, 2011)
- R Maidel, P Shoval, B Shapira, M Taieb-Maiomon, Ontology-content based filtering method for a personalized newspaper. Paper presented at the 2nd conference on recommender systems, Lausanne, Switzerland, 23–25 October 2008.
- L Moulet, O Marino, R Hotte, J-M Labat, A Framework for a competency-driven, multi-viewpoint and evolving learner model. Paper presented at the 9th international conference on intelligent tutoring systems, Montréal, Canada, 23–27 June 2008.
- Open edX: The Open edX MOOC platform (2014), <http://code.edx.org>. Accessed 10 June 2014.
- G Paquette, An ontology and a software framework for competency modeling and management. *Educational Technology and Society, Special Issue on "Advanced Technologies for Life-Long Learning"* **10**(3), 1–21 (2007)
- G Paquette, *Visual knowledge modeling for semantic web technologies: models and ontologies* (IGI Global, Hershey, Pennsylvania (USA), 2010), pp. 302–324
- G Paquette, F Magnan, in *International handbook on information technologies for education and training*, ed. by H-H Adelsberger, Kinshuk, J-M Pawlowski, D Sampson (Springer, Berlin Heidelberg, 2008), pp. 365–405
- G Paquette, O Marino, in *Intelligent and adaptive learning systems: technology-enhanced support for learners and teachers*, ed. by S Graf, F Lin, Kinshuk, R McGreal (IGI Global, Hershey, Pennsylvania (USA), 2011), pp. 213–228
- CA Tomlinson, Mapping a route toward a differentiated instruction. *Educ. Leadersh.* **57**(1), 12 (1999)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com