Competency Comparison Relations for Recommendation in Technology Enhanced Learning Scenarios

Gilbert Paquette, Delia Rogozan, Olga Marino

CICE Research Chair
LICEF Research Center, Télé-université
gilbert.paquette@licef.ca, delia.rogozan@licef.ca,
olga.marino@licef.ca

Abstract. In this paper, we address the problem of competency comparison, providing some heuristics to help match the competencies of users with those involved in task-based scenario components (actors, tasks, resources). Competencies are defined according to a structured competency model based on a domain ontology. We provide a context for recommendation through a learning scenario model. The approach has been implemented by extending an ontology-driven system called TELOS. It has been tested with a learning unit where these comparison relations are used to provide recommendations to users involved in a technology enhanced learning scenario.

Keywords. Adaptivity. Semantic Referencing. User Modeling. Assistance Systems. Recommendation. Personalization.

1 Introduction - The Semantic Adaptive Web

Commercially mature recommender systems have been introduced during recent years in popular e-commerce web sites such as Amazon or eBay. 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 new developments in Web 2.0 and the Semantic Web lead to the idea of an “Adaptive Semantic Web” (Dolog and al 2004) based on the “Web of data” (Heath and Bizer 2011; Allemang D. and Hendler J. (2011) . They open new approaches in the area of recommender systems, in particular for trust-aware recommendation, the use of folksonomies and the ontological filtering of resources (Jannach et al, 2011)

The present contribution addresses some of these issues. It proposes to provide a context for recommendation using a learning scenario model and its implementation through a structure of tasks executed by actors using various kinds of input resources, producing outcomes and interacting with other actors (Paquette 2010). An example for Technology Enhanced Learning is presented in section 2 and used throughout the text to illustrate the main concepts involved here.

We have built an ontology-based competency model, also presented in section 2. It is used for the semantic referencing of actors, tasks and resources in a scenario, and as a basis for recommendation. Unlike other approaches for an ontology-based recommendation, such as OWL-OLM (Denaux et al. 2005) or Personal Reader (Dolog et al., 2004), this competency model extends a domain ontology with mastery levels, e.g. generic skills and performance levels.

In section 3, we describe a method for referencing resources in a learning scenario with such ontology-based competencies. We also address the central problem of competency comparison, providing some heuristics to help match a user's competencies with those possessed by other actors or involved in task or resources in a scenario.

In section 4, we present an application where these comparison relations are used to define recommendation agents, to help personalize a learning scenario. Applications like the one presented here are implemented as an extension of the TELOS ontology-driven system (Paquette and Magnan, 2008), providing a proof of concept of the general approach.

2 Competency referencing of learning scenario components.

2.1 Scenario models for learning contexts

Figure 1 presents a simple scenario model, a screen-shot from our G-MOT scenario editor (Paquette et al., 2011). There are four tasks, two actors (a professor and a student) and some resources that are input to the tasks or produced by the actor responsible (R-link) for the task. Each task 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 task, the student reads the general assignment for the scenario and the list of target competencies he is supposed to acquire. In the second one, he builds a table of planet properties that is validated by the professor, using the information in a PowerPoint document (called *"Planet Properties"*). In the third one, using this table assessed by the professor *("Validated table"),* he compares five properties of planets to find out relations between properties, writing a text on his findings *("Validated relations").* In the last task he is asked to order the 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 recommendation agents have been added to corresponding tasks, in order to provide advice and update the student's competency model with newly acquired competencies. Their action will be explained in section 4.

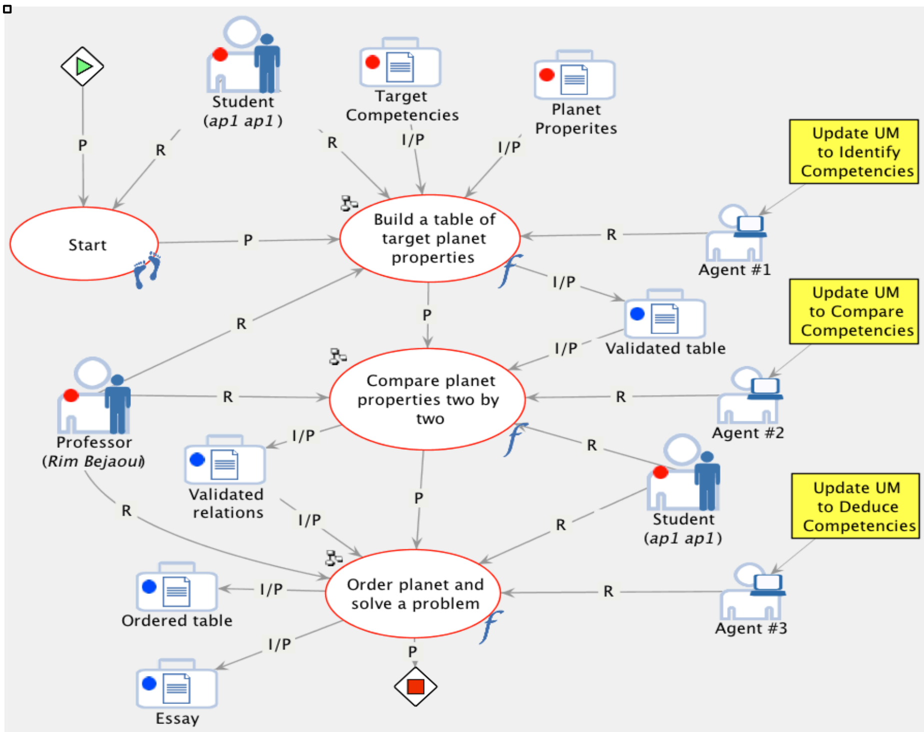

Fig. 1. An example of a scenario model

2.2 Semantic referencing of scenario components

As we have pointed out (Paquette and Marino, 2004), 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 semantic references to scenario components. Two main methods are generally used: semantic references from a domain ontology or natural language statements called prerequisites and learning objectives (as in IMS-LD). Both are not sufficient for our purpose.

In most common practice, unstructured natural language statements from a competency referential are used. Such statements have many problems. First, they are not related to domain ontologies that could describe formally their knowledge part. Second, natural language statements are not appropriate for computation. Computationally, they make it difficult to reference and compare competencies assigned to actors, tasks and resources of a learning scenario. The IEEE-RCD (2007) specification allows optional definition elements as “a structured description that provides a more complete definition of the RCD than the free-form description expressed in the title and description”.

Our competency model corresponds to that goal. It has been published in many conferences, journals and books, and also extensively used in instructional design projects. Devedsic (2006, p.260) describes our model as “a competency structure, corresponding to the domain ontology and represented by entry and target competencies related to the nodes in the instructional structure” (the scenario model).

Unlike other approaches based on ontologies, such as OWL-OLM (Denaux et al. 2005) or Personal Reader (Dolog et al., 2004), the proposed competency model extends a domain ontology with mastery levels, e.g. generic skills and performance levels. In fact, referencing resources with a set of concepts from a domain ontology is an important step, but generally, it is limited to “lightweight ontologies”, i.e. simple taxonomies, thus ignoring the richer structures found in OWL-DL ontologies. Furthermore, to state 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 knowledge. It is a different competency if a user must simply recognize the malfunction of a device, diagnose it or 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.

For that purpose, in our competency model (Paquette 2007, 2010), each competency is defined 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 a combination of performance criteria values. This model can be instanciated to any system of competencies describing them in terms of skills, knowledge and performance, such as the European Qualification

Framework, in which qualifications range from basic level 1 to advanced level 8 (EQF 2012).

This model has been implemented (in a TELOS extension) for referencing actors, tasks and resources in the following way. The domain ontology follows 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 with four performance levels (0.2-aware, 0.4-familiar, 0,6-productive, 0.8-expert), added to the skill level.

For example, using a domain ontology of solar system planets (shown on figure 3) and a competency referential (or model) based on this ontology, competencies can be associated to a resource from the scenario on figure 1. The competencies describing such a resource, (*"Planet Properties")*, could be compared with those of a user (*Gilbert Paquette*) to verify if he has all of them, or some, or none, in his competency model, and offer a recommendation accordingly.

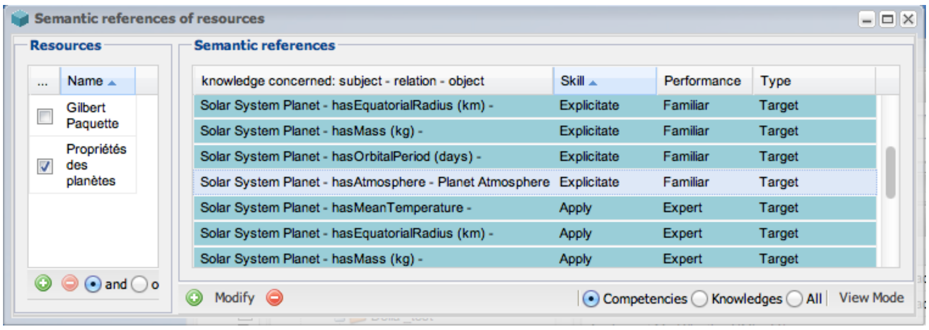

Fig. 2. An example of competency referencing for an actor and a resource

2.3 Tasks, resources and user competency models.

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

A *user competency model* is composed of three main parts (Moulet et al. 2008).

- The core of the model is the 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.
- The competency model contains also documents (texts, exam results, videos, images, applications, etc.) structured into an e-portfolio that presents evidence for the achievement of related competencies.
- The context in which a competency has been achieved is also stored in the model. It includes the date of achievement, the tasks that led to it, the link to the evidence in the e-portfolio and the evaluator of this evidence.

3 Competency Comparison

3.1 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 values associated with the skills and the performance levels.

From a semantic point of view, a recommendation agent evaluates for example if a user's actual competency is very near, near, or far from the prerequisite or target competencies of a resource or a task or from the actual competencies of another user. The agent 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 representation, 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 3.3 we extend the framework to competencies by considering skills and performance .

3.2 Semantic Proximity of the Knowledge Components.

In this section, we focus only on the knowledge part of two competencies to be compared. Maidel et al. (2008), proposes an approach in which a taxonomy is exploited. Five different cases of matches between a concept A in the resource profile and a concept 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 in order, for example, to recommend news to a user according to his preference.

Maidel et al. state that if the use of taxonomy is not considered, the recommendation quality significantly drops. Our thesis is that, for education, taxonomy is not enough either, for only subsumption relations are exploited. We thus propose to define the semantic proximity between knowledge elements, based on their situation in the domain ontology.

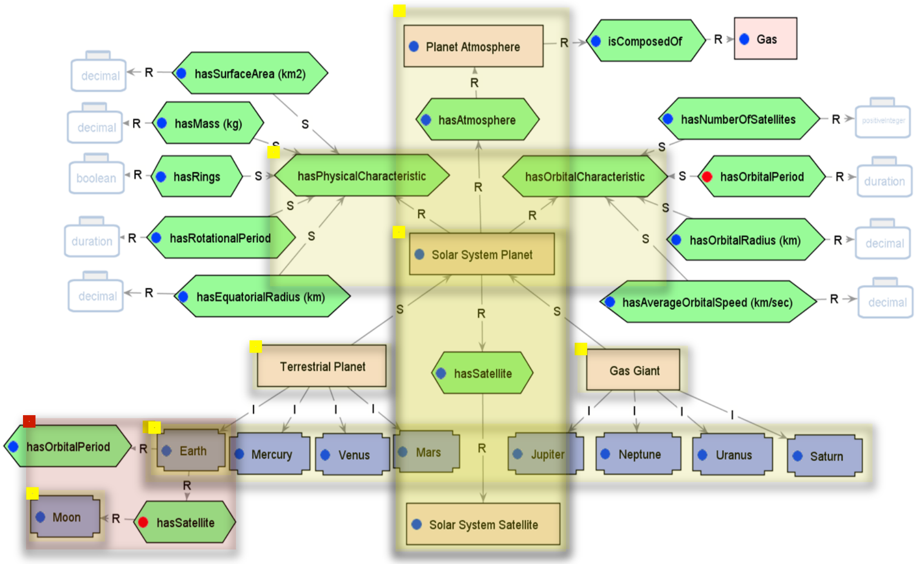

Fig. 3. Domain Ontology on Solar System Planets and some Proximity Relations

Semantic references are components from an OWL-DL ontology that describe the knowledge in a resource. A few examples of these

7

knowledge references are shown on figure 3 that presents part of an ontology for solar system planets[1]. They can take six different forms *solarSystemPlanet* is a class reference (C), *Neptune* is an instance reference, *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'), *solarSystemPlanet/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 6 forms of OWL-DL references to decide on the nearness of two references K1 and K2. 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 subclass or superclass of its domain and range.

Other criteria assert when a reference K_1 is 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, has K_2 as an instance, 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 reference K_2.

3.3 Semantic Relationships Between Competencies.

Let us now extend the comparison between ontology components to add the skill (S) and performance (P) dimensions of the competency model. Figure 4 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 (figure 4). 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 C1 with an analyze skill at an expert level is labeled 5.8 ($S_1 + P_1$). A competency C2 at a level 7.2 or 6.4 will

[1] Unlike other graphic presentation of ontologies, properties are shown as objects (hexagons) between their domain and range classes (rectangles). It this way, the relations between properties are shown on the same graph. Individual are linked to classes by an "I" link.

be considered near and stronger than C1 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, 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 near and stronger and more general than another one.

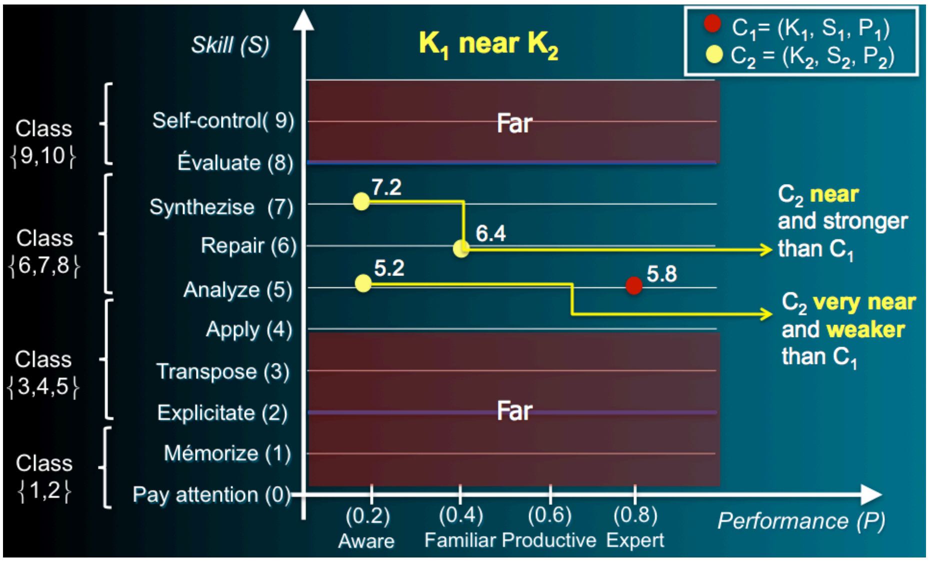

Fig. 4. Comparison criteria for two competencies with their knowledge parts near.

4 Recommendation based on competency comparison

4.1 Competency-based conditions and rules.

Recommendation agents are added to a scenario, linked to some of the tasks called *insertion points,* as in the example of figure 1. The designer defines these agents by a set of rules. In each rule, one and only one of the actors linked to the task at the insertion point is chosen as the receiver of the recommendation. If a triggering event occurs at run time such as "task 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 user has one (or 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/Generic, Near/Specific, VeryNear/Generic, VeryNear/Specific, Stronger, Weaker; or any combination of these relations.
- *ObjectCompetencyList* is the list of prerequisite or target competencies of a task or a resource at/around the insertion point.

Lets take the example of a condition like:

HasAll /NearMoreSpecific / Target competencies for Essay

When it is evaluated, competencies in the user's model are retrieved, together with the list of target competencies for the resource "Essay" The evaluation of the relation "NearMoreSpecific" provides a true or false value according to the method exposed in section 3.3.

4.2 Recommendation actions, an application.

The action part of an agent's rule can perform one or more tasks: give advice to the actor, notify another actor, recommend various learning resources, update the user's model, propose to jump to another task or to another learning scenario.

All these possibilities have been implemented. On figure 1, we have presented a scenario with three recommendation agents. For example, Recommender agent #1 on figure 1 will verify if the student has succeeded the second task in the scenario ("Build a table…"). It has 3 rules, shown on the screen-shot of figure 5.

The rule "Update User Model" transfers the list of target competencies for the task to the student's user model if he has succeeded to build a validated table of planet properties. If he has failed, a second rule will send a notification to the professor to interact with the student. Finally, a third rule provides an advice to the student and recommend consulting a resource shown on figure 5.

Fig. 5. Example of an agent's rule based for updating a user's competency model

5 Conclusion

We have produced an implementation for competency-based assistance that has been tested with a few scenarios and recommendation situations. It provides a proof of concept for the general method. It also provides a workbench to investigate further and extend the methods presented here with variants and a larger range of applications.

First of all, extensive experimental validation will help refine the relation for semantic nearness between OWL-DL references. Adding weights to the various cases would 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.

Our model of multi-actor learning scenarios embeds the idea of collaboration between learners, and between learners and various kinds of facilitators. Recommendation for groups in collaborative scenarios has not been thoroughly explored yet.

Finally, to improve the practical use of approach, some of the task will have to be partly automated and the ergonomics of the system improved. Still, the approach presented here sets the ground for an open and flexible method for semantically aware recommendation systems.

6 References

1. Adomavicus, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6).
2. Allemang D. and Hendler J. (2011) *Semantic Web for the Working Ontologist* (2[nd] Edition), Morgan-Kaufmann, Elevier Inc.
3. Denaux, R., Aroyo, L. and Dimitrova, V. (2005) *An approach for Ontology-based Elicitation of Usr Models to Enable Personalization on the Semantic Web*, Retrieved June 2012, from http://wwwconference.org/www2005/cdrom/docs/p1170.pdf
4. Devedzic, V. (2006) *Semantic Web and Education*. Springer's Integrated Series in Information Systems, 353 pages.
5. Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004). *Towards the adaptive Semantic Web.* Lecture Notes in Computer Science. Berlin, Germany: Springer.
6. Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004). The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies, *AH 2004: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2004. Retrieved June 2012 from:
http://www.kbs.unihannover.de/Arbeiten/Publikationen/2004/ah2004_d_h_n_s.pdf
7. IEEE-RCD (2007). *LTSO Contents – IEE RCD Overview*. http://www.cen-ltso.net/main.aspx?put=264
8. *The European Qualification Framework.* http://ec.europa.eu/education/lifelong-learning-policy/eqf_en.htm. Retrieved August 29, 2012
9. Heath T. and Bizer C. (2011) *Linked Data – Evolving the Web into a Global Data Space.* Synthesis Lectures on The Semantic Web: Theory and Technology, Morgan and Claypool Publishers,
10. Jannack, D., Zanker, M., Felfering A., & Friedrich G. (2011) *Recommender Systems, an Introduction.* Cambridge University Press. New York, USA, 335 pages.
11. Maidel R., Shoval P., Shapira B., & Taieb-Maiomon M. (2008) Evaluation of an Ontology-content based filtering method for a personalized newspaper, *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08),* Lausanne, Switzerland, pp. 91-98
12. Moulet, L., Marino, O., Hotte, R., & Labat, J.- M. (2008). Framework for a competency-driven, multi-viewpoint and evolving learner model. In B. Woolf, et al. (Eds.), *Proceedings ITS 2008, 9th International Conference Intelligent Tutoring Systems,* (LNCS 5091, pp. 702-705). Montréal, Canada: Springer.
13. Paquette, G., Léonard, M. Lundgren-Cayrol, K. (2011) The MOT+ Visual Language for Knowledge-Based Instructional design, in *Instructional Design: Concepts, Methodologies, Tools and Application*. Information Resources Management Association (IRMA), IGI-Global, 2011, pp 697-717
14. Paquette, G. 2010. *Visual Knowledge Modeling for Semantic Web Technology*. IGI Global, 433 pages
15. Paquette, G., & Magnan, F. (2008). An executable model for virtual campus environments. In H. H. Adelsberger, Kinshuk, J. M. Pawlowski, & D. Sampson (Eds.), *International handbook on Information Technologies for education and training, 2nd edition*, (pp. 365-405). Springer
16. Paquette G. and O. Marino (2006), "Learning Objects, Collaborative Learning Designs and Knowledge Representation", in *Technology, Instruction., Cognition and Learning* , Vol. 3, p.85-108, Old City Publishing, Inc.