Paquette, G. (2007). An Ontology and a Software Framework for Competency Modeling and Management. *Educational Technology & Society*, 10 (3), 1-21.

An Ontology and a Software Framework for Competency Modeling and Management

Gilbert Paquette
LICEF Research Center, Télé-université, Montreal, Canada // Tel: (514) 840-2747 ext. 2818
gilbert.paquette@teluq.uqam.ca // www.licef.teluq.uquebec.ca/gp

ABSTRACT

The importance given to competency management is well justified. Acquiring new competencies is the central goal of any education or knowledge management process. Thus, it must be embedded in any software framework as an instructional engineering tool, to inform the runtime environment of the knowledge that is processed by actors, and their situation toward achieving competency-acquisition objectives. We present here some of our results in the last 10 years that have led to an ontology for designing competency-based learning and knowledge management applications. Based on this ontology, we present a software framework for ontology-driven e-learning systems.

Keywords

Ontology-driven e-learning system, Competency acquisition

A search on the Internet is sufficient to show the importance given to competency profiles in human resource management and education. Ministries of education, school boards, and teacher training institutes use competency profiles to define school programs or teachers' required qualities, especially in the use of technologies in education.

Consulting companies present their expertise by enumerating competencies, marketing their services in this way. Other companies offer services or computerized tools to help their prospective customers define or manage the competence of their staff. These services and tools are looked upon as the main asset of an organization from a knowledge management perspective. Governmental agencies or professional associations use competency-based approaches to define conditions to the exercise of a profession and to orient their vocational training programs.

To address the challenges of the knowledge society, we need to better support the process of competency acquisition in the context of lifelong learning, which is more and more required from every citizen. We need more flexible, adaptive learning systems, inside and outside public education systems, before, after, and during work. We need to respond to the huge demand for web-based resources for work and learn to cope with the exponential growth of information, by making the semantic web more and more a reality. We need to make educational modeling more widely used through powerful yet user-friendly tools and methods. Finally, we need to provide citizens with the tools to personalize their learning processes based on evaluation of their competencies.

Explorations in competency management

In this introductory section we will survey a set of applications related to competency management. These applications have been built in various projects we have achieved since 1992. It is important to note that we have started these projects with a clear definition of a competency that has been essentially confirmed and refined by many projects, some of which will be presented in this section.

Our definition is founded on the relation between specific knowledge in an application domain and generic skills. *Competencies are statements that someone, and more generally some resource, can demonstrate the application of a generic skill to some knowledge, with a certain degree of performance.*

For example, suppose we say that a technician can *diagnose the faults in a car engine, for all kinds of car*. This is a competency in which a technician applies the generic skill of *diagnosis* to his knowledge of *faults in a car engine*, with a degree of performance that involves *all kinds of cars*. A lower level of competency would be to *"sometimes identify that there are some faults in a car engine"* because "identify" is a simpler generic skill than "diagnose" and "sometimes" is less demanding than "all the time." These kinds of relationships between competencies are very important for instructional engineering, as we will show later on.

In this section, we will present applications where we have used this definition to help develop and validate the ontology that will be presented in the second section, which gives a more precise meaning to the concept of competency. Then, based on this ontology, we shall define tools and services for a software framework for competency management, which is the subject of third section.

Competency in an Instructional Engineering Method (MISA)

As early as 1992, we started to integrate knowledge and competency modeling within the MISA Instructional Engineering method (Paquette, Crevier, & Aubin, 1994; Paquette, 2002a, 2003). Competency ontologies are central to any instructional engineering methodology. They are important before, during, and after delivery. Before delivery, Competencies serve to guide the design or adaptation of a learning environment. During delivery, they guide the action of facilitators for learners' assistance. After delivery, they help assess the learning results and the evaluation of the quality of a learning environment. We give here two examples of the use of competency modeling for instructional engineering that have served in many of our applications projects.

Guiding the definition of knowledge and activities using competency gaps

Figure 1 presents two of the tools in Atelier distribué d'ingénierie des systèmes d'apprentissage (ADISA)(translated from French as "Distributed Workbench for the Engineering of Learning Systems"), a web-based workbench to support the 35 main tasks of the MISA instructional design method. These tools are used to define a set of competencies associated with a knowledge model that defines the content for a course. In task/tool 212, the designer builds a graph of the knowledge that will be processed by learners and facilitators in an e-learning application, adding "P" labels to the knowledge elements that are important priorities for learning.

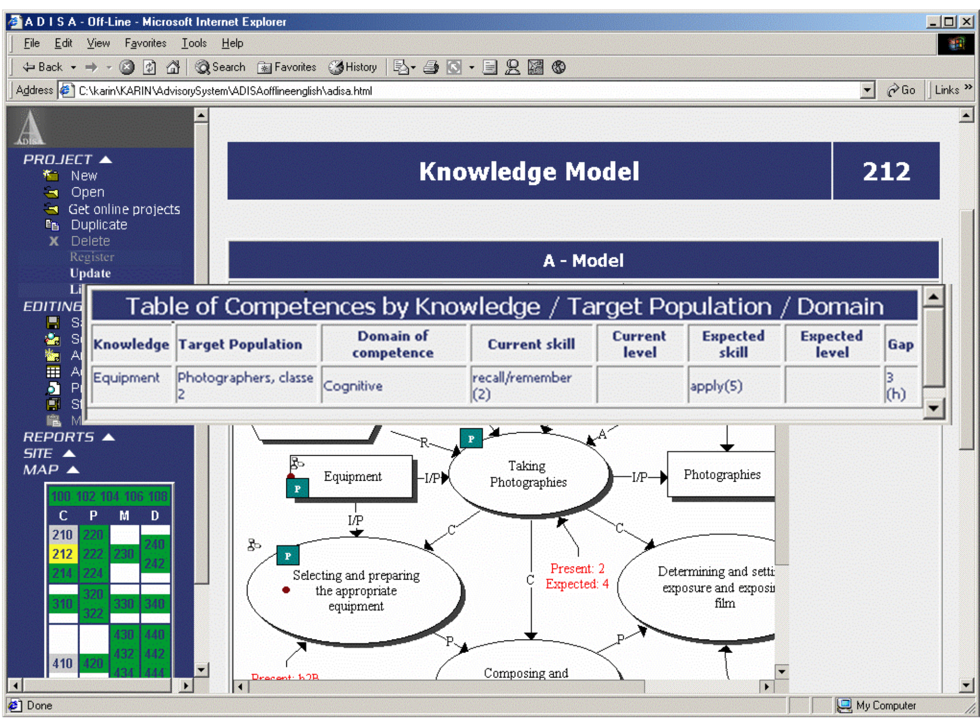

Figure 1. Associating competencies to the knowledge element

In task/tool 214, a list of these priority knowledge units is automatically transferred from the model and displayed to the designer in order for him to assign prerequisite and target competency statements for these knowledge elements. An example for the knowledge of photography, "Equipment," is shown in the figure. For each priority knowledge element and each target population of learners, the skill name and level and the performance name and level are entered for the prerequisite ("current skill" in the figure) and target or expected skill. Here, no performance level is shown: the spaces for current and expected levels are empty.

This data enables the system to calculate the gap between entry and target competency, which is defined here as the absolute value of the difference between the two skill levels. The notion of skill level is related to generic skills' taxonomies that will be explained later in table 4. Here the current or entry skill is "recall/remember," which is the second on a scale of generic skills of increasing complexity. The expected skill is "apply," which is fifth on the same scale. So the gap is simply $| 5 – 2 | = 3$. If the performance level had been defined, we would have been able to obtain a two-dimensional scale, such as the one in figure 8, to evaluate the gap more precisely.

The system can then display a table of the priority knowledge element versus the competency gap (shown on figure 1 for the knowledge entity "Equipment"). Using this table, it is possible to guide the further development of the knowledge model, going back to the modeling tool in task 212. For example, for a knowledge element with gap of 0, it is not necessary to develop the model further; in fact it shouldn't be a learning priority at all. On the contrary, for a knowledge element with a gap of 3 between the prerequisite: recall/remember (level2) and the target: apply (level 5), we should add more related knowledge in the model and thus more activities in the pedagogical model to acquire that knowledge.

Constructing an activity structure based on a generic skill

The generic skills on which a competency is based are processes acting on knowledge in an application domain. They can also be represented as process models (Paquette, 1999). On the left side of figure 2, there is a graph of a generic skill, to simulate a process, with its main operations (ovals) such as "produce examples of the input to the process" to be simulated, "identify the next applicable procedure" in the process, "apply this procedure," and finally "assemble the simulation trace" by collecting the products (rectangles) of these operations. On the graph, four groups of principles (hexagons) are added to constrain the products and/or control the operations of the generic simulation process. Note that this model is totally generic, applicable to any specific knowledge domain, such as Internet processes, manufacturing processes, or others.

The graph on the right side of figure 2 presents a corresponding learning scenario based on a generic skill model in which learners simulate a multimedia production process by performing learning activities that correspond to the main operations in the "simulate a process" skill. Such an activity structure is based on a graph almost isomorphic to the generic process, however, taking a "learning activity" viewpoint. The specific domain vocabulary is used, and the activities are formulated in an "assignment style" format.

This provides the skeleton of a learning design. To complete the process, we need to add resources to help learners achieve their tasks. The important thing here is that the generic process in a target competency provides the structure the learner's assignments. In that way, it is possible to make sure that the learner works at the right skill level, in this case "simulating a process," while at the same time processing a specific knowledge domain.

Constructing a professional training program

We have used the instructional engineering tools and the principles presented above to develop a complete professional training program. In 2003, we were contracted by a Montreal-based company to apply our competency engineering approach and our taxonomy of generic skills, in order to re-engineer a training program offered by the School of the Quebec Bar. This one-year program serves to habilitate every new lawyer in Quebec to professional practice. Working meetings of our team members with an expert committee, composed of 12 experienced lawyers, allowed us to build a relevant knowledge model for the domain of law practice. In a second step, we identified cognitive and socio-affective skills associated with this knowledge, as well as the conditions of performance that are required from novice lawyers to be able to start professional practice.

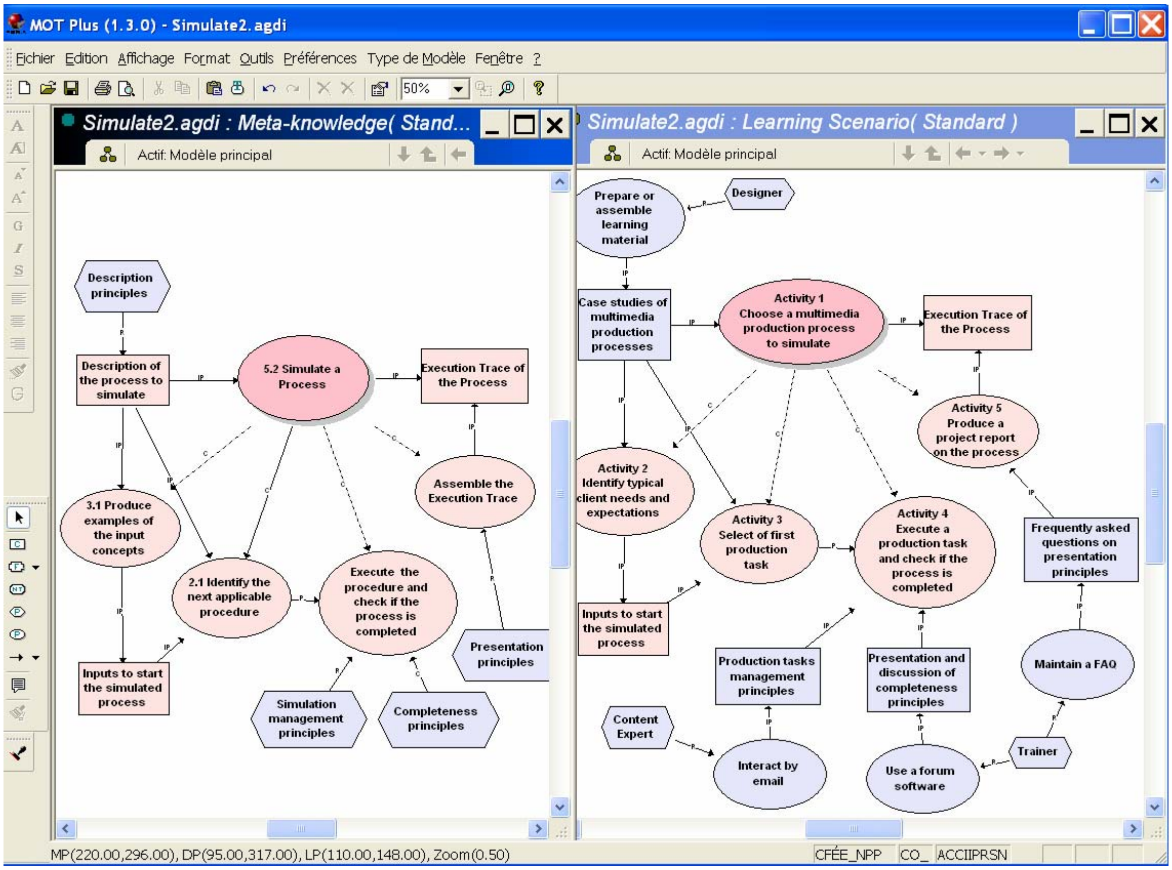

Figure 2. From generic skill to learning scenario definition

Data on these elements was collected during group sessions with the expert committee, and individual face-to-face sessions with some of its members. Questionnaires were filed in by all members. The consultation of different content documents used in the program served to enrich the information. Systematically, the analysis of obtained data led to a document synthesizing a competency profile, which was validated by the Committee and which brought new elements to advance the work and complete the models. The iterative revision of the different versions of the competency profile led to a list of 35 main knowledge elements with their associated target competencies.

Table 1 presents a sample of some of these competencies. Each competency is expressed by a statement specifying the generic skill (in bold italic) that the novice lawyer has to apply to a knowledge element (in italic) according to particular performance conditions expressed in the rest of the competency statement.

Competencies were grouped in five domains based on the knowledge model: A — law concepts, regulations, and standards of the profession; B — communication with the client; C — establishment of a diagnosis; D — elaboration, assessment, and application of a solution; E — management of case data and quality control. The four last categories show that the knowledge model was mainly procedural, describing the main element of law practice as a sound decision for a professional program. The committee attributed a priority to each of the competencies shown in the priority column of table 1.

To plan the new program, it was important to identify, for every competency, the distance between the prerequisite competency that the students should possess before entering the program and the target competency to be acquired by the end of the program. The levels of the target generic skill were first identified by the expert committee, and entry levels were set in the second phase by trainers in the program. The difference between the two is the gap shown in the last column of table 1.

Table 1. A sample of the 35 competencies for the law training program

ID	*Group A — Law concept, regulations, and standards of the profession*	*Priority*	*Entry*	*Gap*
A1	***(6) Analyze*** the *applicable texts of law* to a situation, without help for simple and average complexity situations, with help in complex ones	1	(2)	4
A3	**(3)** ***Specify*** the applicable law regulation autonomously, in any case	2	(1)	2
A8	***(5) Apply*** *pertinent proofs and procedures*, without help for simple and average complexity situations.	1	(2)	3
ID	*Group B — Communication with the client*	*Priority*	*Entry*	*Gap*
B1	***(6) Analyze*** *interactions with the client*, without help in any communication situation.	2	(2)	4
B2	***(9) Evaluate*** the quality of one's capacity *to listen to the client*, without help in any communication situation	2	(1)	8
B4	***(4) Transpose*** in one's social and affective interactions with the client *principles of communication* and *human behaviour*, without help in average complexity situations.	2	(1)	3

Once stabilized, the competencies, their groupings, and the estimated competency gap contributed to define the structure of the new program. As shown in table 2, competencies were distributed in learning units (called courses), taking in account groupings, priorities, and the competency gap between entry and target levels.

The gap between the entry and target competency levels proved to be very important for the construction of the program. There were 9 competencies with a gap from 1 to 3, 18 with a gap of 4 or 5; 5 with a gap of 6 or 7; and 5 with a gap of 8. The competencies were distributed in a spiral approach into four sequential learning units (courses 1 to 4) according to the gap. For example, the B2, and E4 to E7 competencies (with a gap of 8) were integrated in all four courses to increase progressively the generic skill and performance levels of the learners. Competencies A1 and others were distributed only in the first two courses because they are easier to acquire. Competencies A3 and others were included only in the first learning unit, which seems to be sufficient for their acquisition. The target competencies in the courses serve as learning objectives to be measured by exams and other means of evaluation. The following phases of the project have focused on building learning scenarios for each learning unit or sub-unit based on the generic skills in the associated competencies.

Table 2. Distribution of competencies into courses of the program

Gap	Competencies	Course 1	Course 2	Course 3	Course 4
1-3	A3, A4, A5, A6, A7, A8, B4, C2, C6	x			
4-5	A1, A2, A9, A10, B1, B3, C1, C3, C5, D2, D5, D6, E1, E2, E3, E8	x	x		
6-7	C4, D1, D3, D4, E9	x	x	x	
8	B2, E4, E5, E6, E7	x	x	x	x

Integration in an LCMS (Explor@-2)

The same general approach to competency modeling has been integrated in an e-learning delivery system called Explor@ (Paquette & Marino, 2005). Unlike most LCMS, Explor@ is built around two structures: the activity structure (or learning design), which breaks down a program or course into smaller activity structures, activities, and resources, and the knowledge/competency structure, which presents a hierarchy of concepts in an application domain, with their associated entry and target competencies. Figure 3 presents the tools that help create and manage these two structures.

The large window on the left presents a tree of concepts from a lightweight ontology in a subject-matter domain (in this case, eco-agriculture). The nodes of this tree are concepts from the domain ontology and the leaves serve to select a generic skill associated to its parent node. When selecting a leaf like "Analyze-6" with parent node "agriculture practice," the right part of this window enables a designer to document corresponding competencies, while writing the target competency statement. The knowledge and skill elements are copied automatically from the tree structure. The designer then adds the meta-domain (here "cognitive," but this could also be "affective" or "psycho-motor"). The performance level is set by selecting some or all of the "performance criteria" in an auxiliary window, criteria that are combined by the system into A, B, C, or D level (here B-Familiarity). The rest of the right side of the window is to fix similar elements for the entry competency on the same knowledge, based here on the "Apply-5" generic skill.

The lower right side of the window serves to associate these competencies with activities and resources in the activity structure shown in the other window. This is done by simply selecting activities or resources from this window; resources are on the leaves of this tree below the activity or activity structure where they are used or produced.

Another auxiliary tool (not shown on figure 3) displays a summary table of competencies at different levels of the activity structure, enabling designers to distribute relative weights to compose a global evaluation. From this data, it is possible to generate automatically a self-diagnosis questionnaire like the one in figure 4 below, where learners can self-assess their competencies and select proper activities and resources to help improve them. Similar interfaces can be generated for trainers to help them evaluate the students' progress.

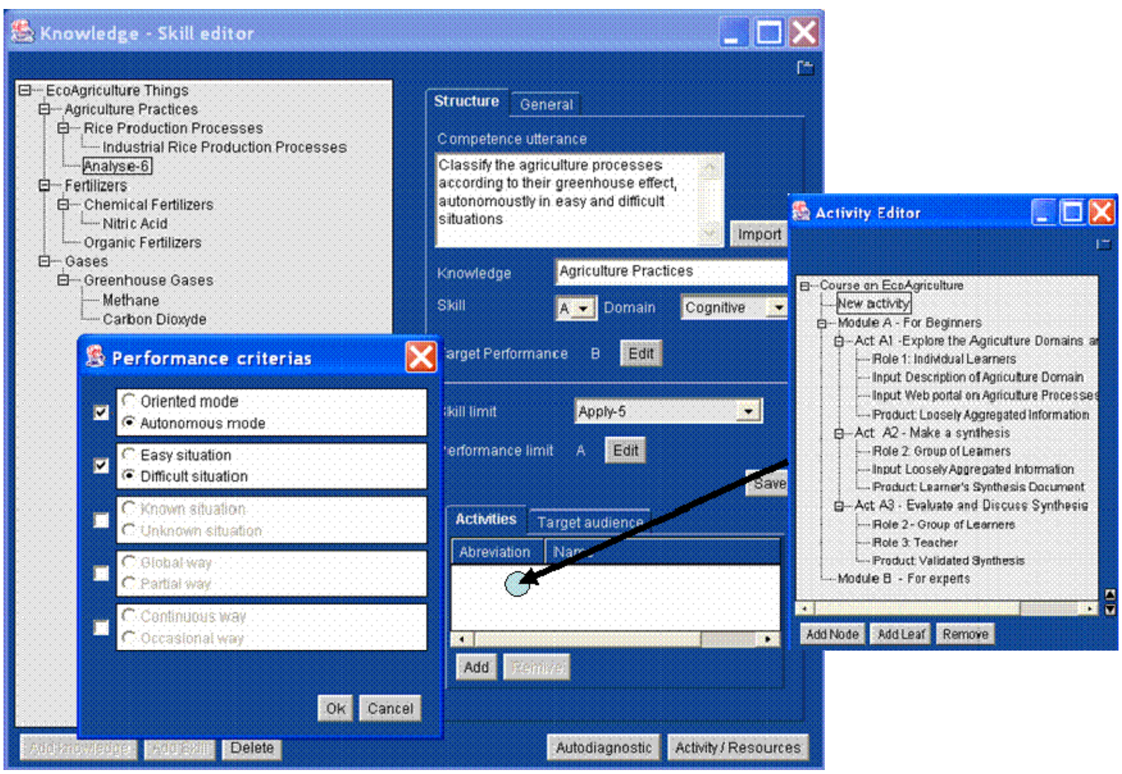

Figure 3. Competency management in Explor@

Emergent self-composed training programs for life-long learning

Figure 4 shows screens of a user-friendly self-diagnosis web tool that can help students diagnose their competency and compose their own training programs with or without the help of a trainer or facilitator (Ruelland, Brisebois, & Paquette, 2005). The tool comprises three steps displayed in the three windows of the figure.

On the first page, a list of competencies imported from a competency editor is presented to the user (learner or trainer). For each competency, the user selects his/her actual performance level from among four levels.

On the second page, a global summary of these combined levels is displayed in the form of a bar graph of the gaps between actual and target competencies, to identify strengths and weaknesses.

On the third page, recommendations for a plan of action are provided in the form of resources associated with each competency or competency group. Access is given to these resources through a hyperlink to enable user navigation between the resources. The resources can be any document, website, online or blended learning course, or address of a resource person that have been previously associated with one or more competencies. The association tools between competencies and resources can be made available to students as well as trainers to enable the composition of a training program for lifelong learning. In that case, federated search tools in learning object repositories can be used to find appropriate resources and link them to competencies.

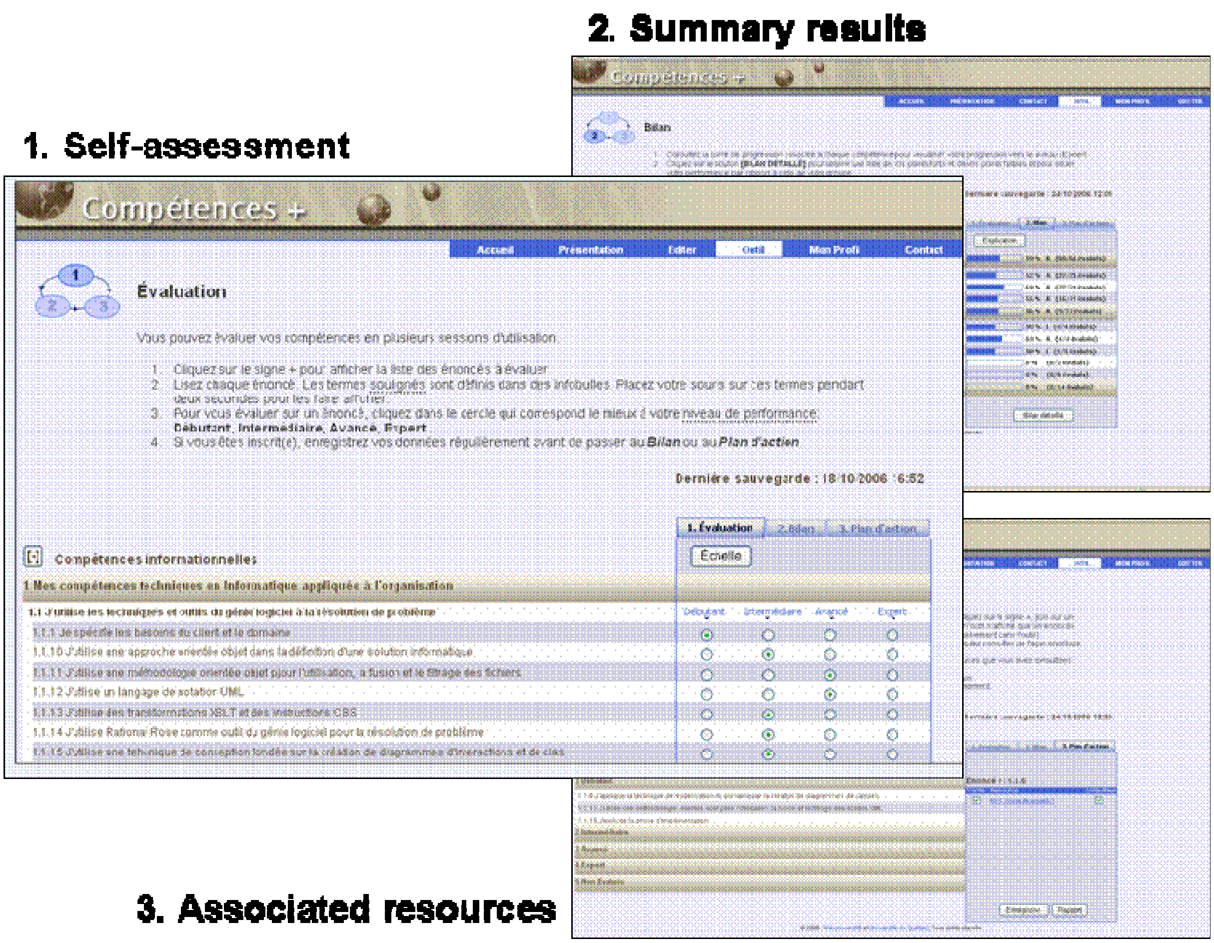

Figure 4. A competency self-assessment tool

An ontology for competency modeling

Drawing on the previous work and the experience gained in numerous projects, we now present an ontology for competency modeling that combines the concepts of knowledge, skill, attitudes, and performance. It is rooted in different fields of research such as instructional design, software engineering, and artificial intelligence. It provides ways to annotate semantically resources in e-learning and knowledge management environments, in particular to define competencies of individual actors, prerequisites and goals for activities and resource content, evaluation criteria, and personalization capabilities for e-learning and knowledge management applications.

Beyond the textual competency statement

Most often, competencies are expressed as simple, plain-language sentences, stating informally that a group or person has the capacity or the knowledge to do certain things. Competency profiles are, in general, loosely structured collections of such texts that are not always easy to interpret, communicate, or use, especially if the goal is to plan learning events to support the acquisition of new competencies.

Efforts are being made to facilitate the use of competencies in education and training. For example, the IMS organization, involved in defining e-learning standards, produced in 2002 a specification for a Reusable Definition of Competency or Educational Objective (IMS-RDCEO 2002). It defines an information model for describing, referencing, and exchanging definitions of competencies, primarily in the context of online and distributed learning. Its goal is to enable the interoperability among learning systems that deal with competency information by providing a means for them to refer to common definitions with common meanings.

As stated in this RDCEO documents, "the word competency is used in a very general sense that includes skills, knowledge, tasks, and learning outcomes." Furthermore, "the core information in a RDCEO is an unstructured textual definition of the competency that can be referenced through a globally unique identifier." The RDCEO does not provide any structured model for a competency, but it mentions that "this information may be refined using a user-defined model of the structure of a competency."

As we pointed earlier (Paquette & Rosca, 2004; Paquette & Marino, 2005), a crucial area where a learning design specification like IMS-LD needs to be improved is knowledge representation. Actually, the only way to describe the knowledge involved in a learning design is to assign *optional* educational objectives and prerequisites to the unit of learning as a whole and/or to some of the learning activities.

Without a structural model for a competency, these entry and target competencies will be unstructured pieces of text, which are difficult to use in e-learning systems, forbidding, for example, consistency checking between different levels of the LD structure, and even at the same level between the content of learning activities, their related resources, and the learners' competency. In fact, in IMS-LD the knowledge in learning resources is not described at all, and the actors' knowledge and competencies are indirectly defined by their participation in learning units or activities only if educational objectives are associated.

Without a good representation of the knowledge and competency to be processed, a delivery system will be unable to help its users according to their present and expected competency state. In other words, the system will be unable to provide personalized learning activities for its users. What we need is both a qualitative structural representation of knowledge in activities and resources, but also a quantitative one providing a metric for evaluating competency gaps during the learning process. The association between learning objects (documents, tools, actors, activities) within a unit of learning, and the knowledge and competencies they possess, contain, or process is a key concept for semantic web application (Berners-Lee, Hendler, & Lassila, 2001).

The knowledge domain in an e-learning or knowledge management application can be structured in many ways: dictionaries, thesaurus, book summary, library catalog, indexes and metadata, knowledge graphs, ontologies, etc. The tree organization of a knowledge domain is an important property that can significantly reduce the processing, but it is insufficient to describe the rich network of relations that ties the concept structures. It needs to be complemented with relations betweens concepts and axioms in order to sustain more refined mechanism of conceptual matching and inference. In other works, we need to use some form of domain ontology (Davies, van Harmelan, & Fensel, 2002, Breuker, Muntjewerff, & Bredewej, 1999).

But if we use only domain ontologies without defining mastery levels for the knowledge, we limit ourselves to weak semantic management capabilities, both by human facilitators and computer support systems. We can use different mastering scales: simple quantitative percentage, levels in Bloom taxonomy, and combinations between generic skills taxonomies and performance levels, which is our proposal. The description of knowledge mastery must be reasonably simple to be manageable. Still, the levels must correspond to clearly identified generic processes such as applying, synthesizing, or evaluating knowledge.

Combining the preceding requirements suggests that a good candidate for the semantic indexing of educational resources, actors, and activities will be a combination between domain ontologies and a simple and expressive generic skills ontology.

A competency ontology

The basis for the competency ontology we will now introduce can be found in many related fields, such as mathematical logic (Thayse, 1988), science methodology (Popper, 1967), problem solving and its teaching (Polya, 1957), educational technology (Romiszowski, 1981 Merrill, 1994), software and cognitive engineering (Breuker & Van de Velde, 1994; Steels, 1990), and artificial intelligence (Pitrat, 1991).

Our definition is founded on the relation between specific knowledge and generic skills. In his work on artificial intelligence, Jacques Pitrat (1991) produced an important synthesis in which he distinguishes several meta-knowledge categories and proposes the following definition: "Meta-knowledge is knowledge about knowledge, rather than knowledge from a specific domain such as mathematics, medicine or geology."

In the education sciences, Romiszowki (1981) expresses very well the simultaneous phenomenon of knowledge acquisition in a particular domain and the meta-knowledge building of generic skills: "The learner follows two kinds of objectives at the same time — learning specific new knowledge and learning to better analyze what he already knows, to restructure knowledge, to validate new ideas and formulate new knowledge." The same idea is expressed by Pitrat: "Meta-knowledge [generic skills] is being created at the same time as knowledge." In other words, meta-knowledge or generic skills develop while they are applied to knowledge in a particular field. Anybody learning new knowledge uses generic skills (at least minimally) without necessarily being aware of it. However, using generic skills should really be a learner's conscious act.

On a more practical ground, we studied (Paquette, 2002b) a sample of competency profiles in diverse sectors including large companies (such as Hydro-Quebec), the Public Service Commission of Canada, professional requirements, high school curricula, libraries, and information processing. We find a large diversity of goals, uses, and even explicit or implicit conceptual framework differences. Our goal is to find unity behind that diversity to be able to extract models of competency that can be used at an operational e-learning framework.

Competencies are statements that link together skills and attitudes to knowledge required from a group of persons and, more generally, from resources. Some examples integrate elements apart from these or ignore some of them. For example, the competency profile of the Public Service Commission of Canada presents a model that lies partly outside the realm of the competency definition, taking in account the interests and beliefs of the public servants. At the other end, statements like "ability to plan work," "openness to critique," and "some knowledge of creation tools" in a competency profile for multimedia producers seem insufficiently precise, describing respectively a generic skill, an attitude, and a knowledge element that are part of a competence but insufficient to describe it.

In most competency profile definitions however, a competency is defined as a combination of skills, attitudes, and knowledge that enable a group or person to fulfill a role in an organization or society, for example, to act as a lawyer, nurse, technician, teacher, information seeker, media producer, or any other profession. Such roles require general competencies that might be shared with other roles, as well as more specific competencies required by specific task, context, and problems they will have to solve. For example, the role of a nurse will not be the same in a developing country as in a society where there exists a highly organized, sometimes bureaucratic, health system.

Figure 5 presents the top level of the ontology that we propose, defining the main competency concepts that will be explained and expanded upon later on.

Competencies serve to annotate resources, human as well as media resources, giving them a semantic meaning as to the knowledge and skills they own or contain. These annotations can represent prerequisites to achieve a task, or to be attained as a result of a task. They can also be declared as actually owned by the resource they annotate. Each competency is composed of a single competency statement, exactly one generic skill that may require precision using performance indicators, and at least one knowledge entity. The competency statement is a plain-language phrase that refers to the other components, stating that the generic skill (with optional performance indicators) can be applied to the knowledge.

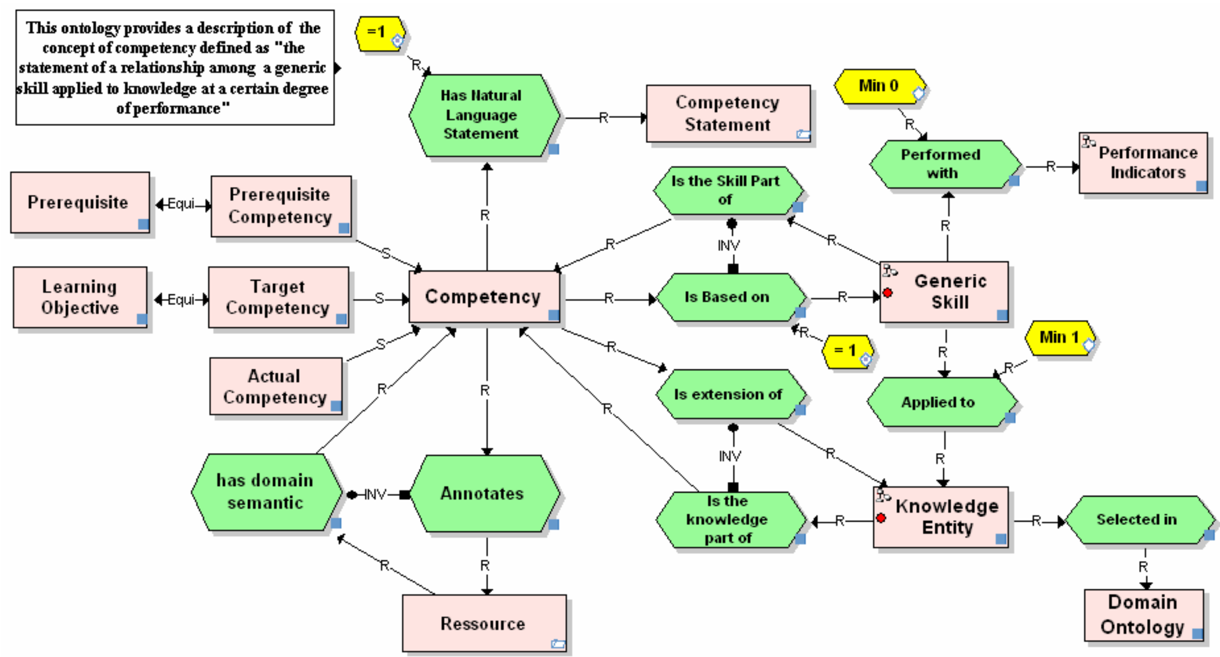

Figure 5. Top-level ontology for competency definition

[*Note.* This graph uses MOT + OWL graphic syntax that covers all OWL-DL primitives (W3C 2004). Rectangles represent classes, and hexagons represent properties linked to their domain and co-domain by incoming or outgoing R links. S links from one class to another means that the first is a sub-class of the second.]

The knowledge part of the competency can be a concept, a procedure, a principle or a fact that is selected in a domain ontology. In a competency profile for a profession such as nursing, this knowledge part will be selected in a healthcare-structured description of facts, concepts, procedures, or principles. In a competency profile for media producers, the knowledge entity will be one of the techniques, methods, objects, or products from the multimedia domain. In general, we will consider that the competency ontology is extended by an application domain ontology from which the knowledge part has been selected as a class (concept) or an individual of the ontology.

A generic skill is a process that can be applied to knowledge in more than one application domain, for example to perceive, memorize, assimilate, analyze, synthesize, or evaluate knowledge items. A generic skill is described by an action verb, sometimes with performance indicators such as "in new situations" or "without help" that serve to make the skill more specific, while remaining independent from any application domain. For example, a generic skill like "establish a diagnosis," or "establish a diagnosis in new situations without help" can be applied in diverse application domains to knowledge items such as "skull fracture," "car motor failure," or "exam failure risk." A generic skill is also selected from a generic skill's ontology, which is considered an extension of the competency ontology.

Table 3 gives a sample of competencies from different sources showing how this competency ontology can be applied. Note that the fourth and fifth examples are merely expressions of a generic skill without a knowledge part, so we have concluded that implicitly it was meant to cover all knowledge in the application domain ontology. The fifth one is in fact two competencies that should be separated because a competency contains only one generic skill. Note also that the second and the fifth competencies have a performance indicator, respectively "accurately" and "in everyday life."

Table 3. A sample of competency statements in different fields and their breakdown

Source	Competency Statement	Generic Skill	Knowledge Entity
ANCI profile for nurses (1)	Demonstrates knowledge of legislation and common law pertinent to nursing practice	Apply	Australian law related to nursing practice
ANCI profile for nurses (1)	Analyzes and interprets data accurately	Analyze without error	Patient healthcare data
Multimedia Producer (2)	Ability to evaluate project's feasibility	Evaluate	Project description
Multimedia Producer (2)	Ability to convince others	Influence	Team members and clients
MEQ — Student Competencies (3)	Analysis and synthesis capability	Analyze Synthesize	All subject matter in the curricula
MEQ — Student Competencies (3)	Apply in everyday life, rules of life in society	Apply in everyday life	Rules of life in society
Teaching Competencies (4)	Operates within the framework of law and regulation	Apply	Laws regulating the teaching profession
Teaching Competencies (4)	Plans purposeful programs to achieve specific learning outcomes	Synthesize	Programs addressing specific outcomes
Information Literacy Profile(5)	Identifies a variety of types and formats of potential sources for information	Identify	Types and formats of information sources
Information Literacy Profile(5)	Determines whether the initial query should be revised	Evaluate	Query for information

Extended examples of these competency profiles can be found in Paquette (2002 b):

Generic skills sub-ontology

We will now expand the competency ontology for the generic skill component of a competency requirement. The backbone of this sub-ontology is a generic skill taxonomy that presents more and more specialized classes of intellectual processes (such as memorize, transpose, analyze, or evaluate) that can be applied in different knowledge domains. Each generic skill class groups individual knowledge-processing activities.

Possessing a generic skill means that a learner can solve a corresponding class of problems (Chandrasekaran, 1987; McDermott, 1988; Steels, 1990). For example, if a learner possesses a diagnostic or classification skill, it implies that this learner is able to solve some diagnostic or classification problems with a certain degree of proficiency. Another view is to see cognitive skills as active, procedural meta-knowledge (generic processes) that can be applied to knowledge (Pitrat, 1991) A third view considers the association between cognitive skills and application knowledge as objects to be learned together, such as educational objectives (Bloom, 1975; Krathwohl, Bloom, & Masia, 1964; Reigeluth, 1983; Martin & Briggs, 1986). Integrating all three viewpoints gives us a solid foundation for a generic skills taxonomy.

Basic hierarchy of generic skills. Table 4 presents an overview of the proposed skills taxonomy. This table displays a loose correspondence between an artificial intelligence taxonomy (Pitrat, 1991), a software engineering taxonomy (Breuker & Van de Velde, 1994; Schreiber, Wielinga, & Breuker, 1993), and two educational taxonomies (Bloom, 1975; Romiszowski, 1981). Although the terms are not in direct correspondence, table 2 distributes them into 10 levels that lay the foundation for our taxonomy (Paquette, 1999; 2003), shown in the left part of table 2. It portrays three layers, from left to right and from generic to specific. It could be expanded to more levels for additional precision.

We will now discuss some of the properties of this taxonomy. Contrary to the behaviourist view on learning objectives, intellectual skills are viewed here from a cognitivist viewpoint as processes that can be described, analyzed, and evaluated by themselves or in relation to various knowledge domains.

Table 4. Generic Skills Processes compared to processes in other taxonomies

Generic Skills Classes			Active meta-knowledge (Pitrat)	Generic problems (KADS)	Cognitive objectives (Bloom)	Skills cycle (Romiszowski)
1	2	3				
Receive	**1. Acknowledge**					Attention
	2. Integrate	2.1 Identify 2.2 Memorize			Memorize	Perceptual acuteness and discrimination
Reproduce	**3. Instantiate/Specify**	3.1 Illustrate 3.2 Discriminate 3.3 Explain	Knowledge Search and Storage		Under-stand	Interpretation
	4. Transpose/ Translate					Procedure Recall Schema Recall
	5. Apply	5.1 Use 5.2 Simulate	Knowledge Use, Expression		Apply	
Produce/Create	**6. Analyze**	6.1 Deduce 6.2 Classify 6.3 Predict 6.4 Diagnose	Knowledge Discovery	Prediction, Supervision, Classification, Diagnosis	Analyze	Analysis
	7. Repair			Repair		Synthesis
	8. Synthesize	8.1 Induce 8.2 Plan 8.3 Model/Construct		Planning, Design, Modeling	Synthesize	
Self-manage	**9. Evaluate**		Knowledge Acquisition		Evaluate	Evaluation
	10. Self- control	10.1 Initiate/Influence 10.2 Adapt/Control				Initiation, Continuation, Control

Table 5 presents process definitions to compare generic skills in the first layer. For each class of generic skills, it shows input and products, as well as examples in the class. These definitions support the hypothesis that skills in the first layer are ordered from simple to complex. Self-management skills involve explicit meta-cognitive operations for evaluation and decisions; this entails the need for production/creation operations to be performed. Creation or production of new knowledge from more specialized knowledge entails the use of reproduction skills. Reproduction skills are essentially instantiation or translation from more general knowledge that requires reception skills. Finally, reception skills involve only attention and memory operations that are needed in reproductive processes.

Generic skill's complexity. We provide here a definition of a generic skill's complexity: Skill A is more complex than skill B if for any generic process P in class A, there exists a generic process in class B acting as a sub-process of P.

We have provided elsewhere (Paquette, 2002b) evidence of a complexity ordering for the 10 classes in the second layer of our taxonomy by constructing process graphs of these generic skills. For example, the "simulate" skill (level of complexity 5.2) can be broken down into four sub-processes: produce examples of the input concept (instantiate: level 3), identify the next applicable procedure (identify: level 2), execute the procedure applying its control principles (apply: level 5, in a simpler way), and finally assemble the simulation trace (transpose: level 4). This shows, according to our definition of complexity, that "simulate" is more complex than skills at levels 2, 3, and 4.

Meta-domain. Generic skills, being processes acting on knowledge, can be classified according to the kind of input they process or produce. For example, "to persevere and to adapt a course of action in a project" is a process in which affective and social input and output are essential. In Paquette, 2002b, we built a complete table showing examples in the cognitive, affective, social, and psycho-motor meta-domains for each of the 10 major skills in the second layer of the taxonomy. This shows that this taxonomy can be interpreted in each of the four meta-domains: cognitive, psycho-motor, affective, and social. For example, we can repair theories and movements, as well as attitudes or social relations. What differentiate these four meta-domains is essentially the type of input to a skill and its resulting outputs. If the stimuli or the result concerns rational thought, motor capacities, affective attitude, or social interactions, we will label the skill to be cognitive, psychomotor, affective, or social, respectively.

Table 5. Comparison of generic skills, from simple to complex (first layer)

Name of skill	Definition	Examples
Receive	Input = internal or external stimulus; Product = facts or knowledge found or stored in memory	Pay attention to an event, to a movement, to an emotion, to a social context; Identify knowledge, associated impressions; Memorize knowledge, impressions.
Reproduce	Input = knowledge and models; Products = facts obtained through instancing or knowledge obtained through reformulation	Use examples to explain or illustrate a concept, a procedure, or a principle; Use a model to explain facts; Simulate a process.
Produce/ Create	Input = knowledge and models; Products = new knowledge or models resulting from analysis and synthesis	Classify objects according to a taxonomy; Repair defective system components ; Plan a project; Model and build a system.
Self-manage	Input = knowledge, models, values; Product = knowledge, models, meta-knowledge (values or generic skills) linked to domain model	Assess knowledge validity or self competence; Initiate a change process after assessing the situation; Apply a generic strategy to improve learning and performance.

Sub-ontology for generic skills. Figure 6 summarizes the preceding discussions on the generic skill's properties, presenting the resulting sub-ontology for generic skills.

The taxonomy is ordered by layers (using specialization S links) from left to right and from general to more specialized skills. The first layer shows a "C" property between the four classes of skills. This property reads from bottom to top as "is more complex than." We add here a transitivity axiom for the C property by adding an arrow label on its graphic symbol. This property also exists between skills of the second layer. Only some of C property symbols are shown in figure 6 to ease the graph's readability, but there is one, for example, from synthesize to repair and from repair to analyze. On the other hand, there are no direct C properties between skills of the third layer, only through their second-layer parent classes.

This MOT + OWL graph shows a third type of OWL objects representing individuals. It serves to assert that generic skills have a meta-domain property that can have as a value "cognitive," "affective," "social," or "psycho-motor", as well as any combination of these values (the "max 4" cardinality axiom affecting the property). It also shows that, based on the second layer's total ordering by complexity, we can assign one and only one number from 1 to 10, representing the complexity level of a generic skill. In OWL terminology, it means that the "has skill level" property is functional. A corresponding axiom is added by a label on the hexagon representing this property.

Performance indicators and levels

We now complete the competency ontology by developing a sub-model for the "performance indicator" class shown in figure 5. There are many possible performance indicators that are used by practitioners and some that we have explored in our own projects in the last 10 years. The ones shown in figure 7 have been found most frequently useful.

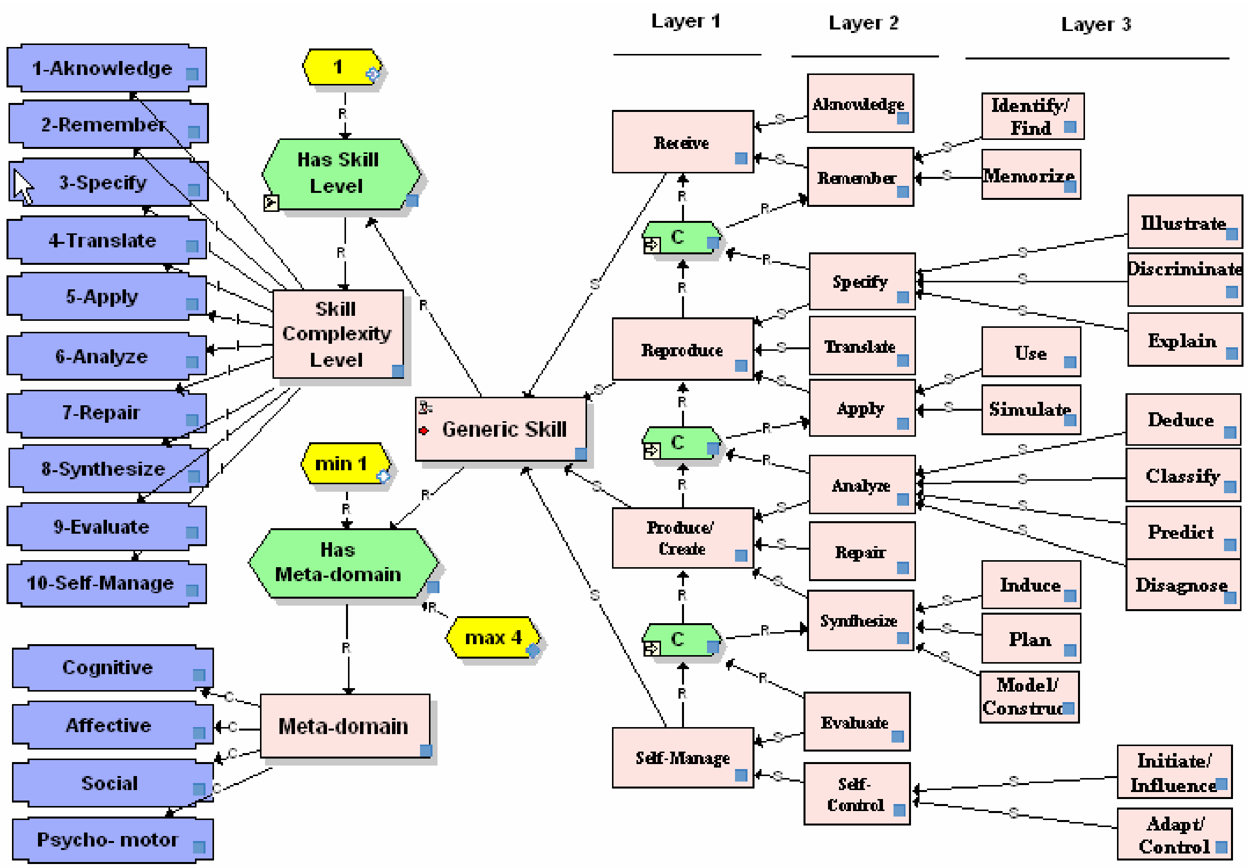

Figure 6. Extension of the competency ontology to generic skills

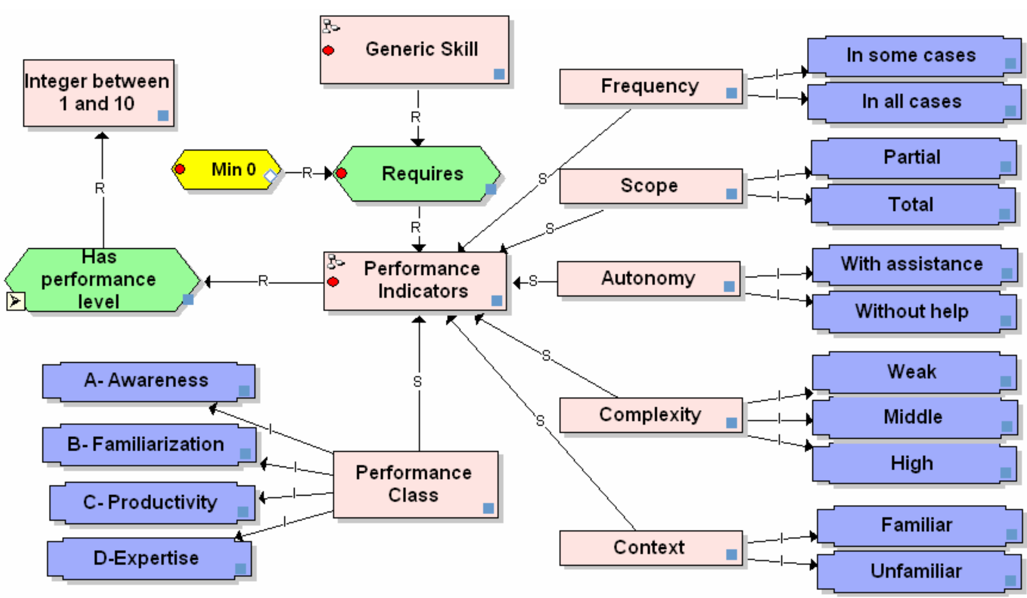

Figure 7. Extension of the competency ontology to performance indicators

For any generic skill, it is possible to add performance indicators such as frequency, scope, autonomy, complexity and/or context of the use. For example, a competency like “diagnose the source of malfunction of a car engine” could be made more precise by adding at the end performance indicators such as “in all cases” or “in the majority of cases” (frequency), “for part of the causes” or “for all causes” (scope), “without help” or “with little assistance” (autonomy), “for high complexity engines” (complexity), or “in unfamiliar cases” (context of use). Some of these values are shown in figure 7 as instances of the ontology. Other individuals and other values could be added to extend the ontology. The usefulness of such indicators is to help build ways to assess the competency, for example, to design exam questions or to register student actions in some model of his or her progress.

Table 6. Performance Categories or Levels vs. Other Indicators

CRITERIA	PERFORMANCE LEVELS			
	Awareness 0.0 – 2.5	Familiarity 2.5 – 5.0	Mastery 5.0 – 7.5	Expertise 7.5 - 10
Frequency	Sometimes	Always	Always	Always
Scope	Partial	Partial	Partial	Total
Autonomy	With assistance	With assistance	Without help	Without help
Complexity	Weak	Weak	Middle	High
Context	Familiar	Familiar	Familiar	Unfamiliar

Alternative and more simple performance indicators classify performance for a generic skill in four broad categories: “awareness,” “familiarization,” “productivity,” or “expertise,” or simply by a number on a 1–10 scale for the performance level. These categories or levels can be direct evaluation results, or they can be calculated from the other indicators. One way to combine indicators or criteria to define performance classes or levels is shown in table 6.

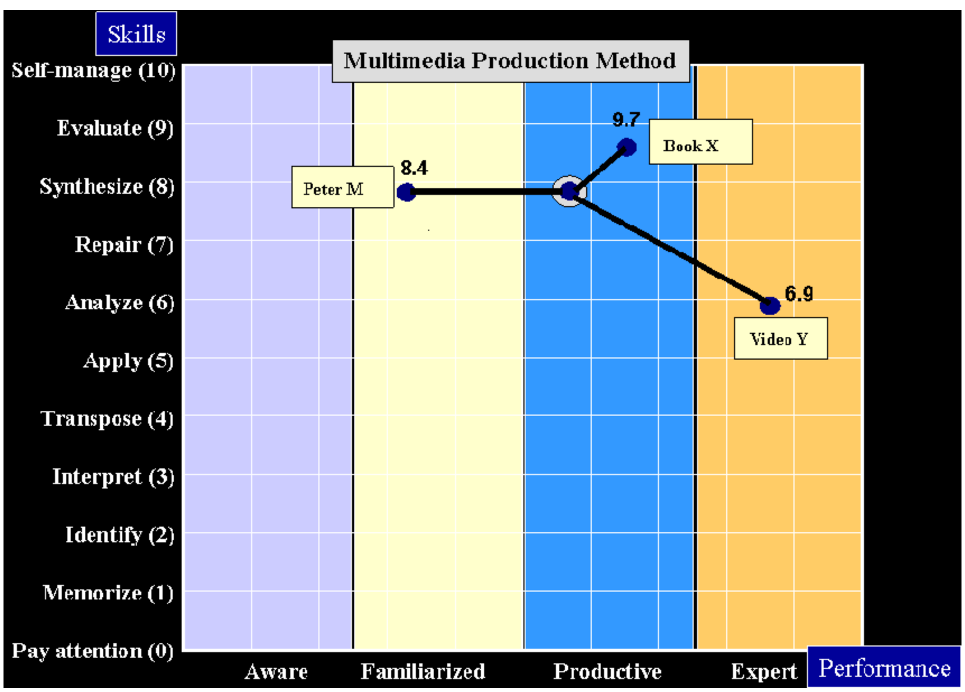

Figure 8. Situating resources on a skills/performance scale

Competency scale

By combining the generic skills' levels with performance levels, we can design a two-dimensional competency-based scale that will help situate resources according to their competency for a certain knowledge item. For example, figure 8 shows such a competency scale for the knowledge of a "multimedia production method." It shows a course having a target competency of 8.6, which means it aims to "synthesize productively a multimedia production method." For that course, Peter M has an actual competency of 8.4, which means he is "familiar with synthesizing a multimedia production method." Video Y, at a level of 6.9, may not be very useful for that course, except maybe as a refresher, because it focuses on "analyzing at expert level a multimedia production method," which is at a lower generic skill level.

A software framework for competency management in TELOS

This concluding section will summarize part of the actual research we are conducting within the LORNET research network to generalize and integrate the concepts presented above into the architecture of the TelELearning Operating System (TELOS).

TELOS: An ontology-driven architecture

TELOS is basically an assembly-and-coordination system. The term "TelE-Learning Operating System" means that TELOS is planned essentially as a set of coordination and synchronization functionalities supporting the interactions of persons and computerized resources that compose a learning or knowledge management system. It integrates human and computer agents using two basic processes: semantic representation of resources and resource aggregation.

An important goal is to help the system survive the rapid evolution of technologies by embedding in the system technology-independent models expressed as ontologies. The TELOS system is able to reuse ontologies as "conceptual programs." In this vision, the conceptual models are not just prerequisite to building the TELOS system; they are part of the system and its most fundamental layer.

Global view of the TELOS competency management framework

The TELOS Competency management framework is composed of the 10 major tools shown in figure 9. The tools in the first column serve to design fundamental elements of a TELOS Learning and Knowledge Management Application (LKMA).

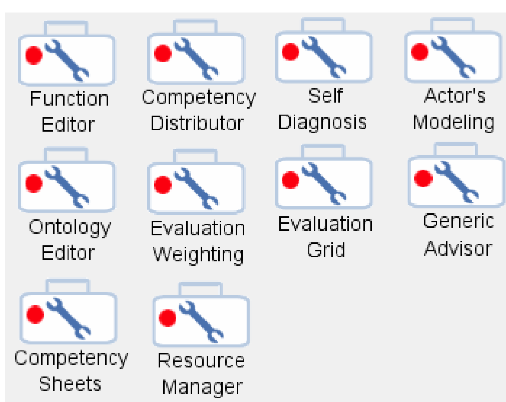

Figure 9. TELOS Competency Framework

A Scenario Editor defines the flow of control and data between actors, activities, and the resources used and produced by the actors in the activities of learning design or workflow. In particular, the graphic design can be translated in an XML file compliant with the IMS-LD specification (2003).

The Ontology Editor defines a domain semantic structuring as the main concepts, properties, and individuals to be studied in an application domain. It is complemented by a Competency Sheets tool that adds competency definitions to some of the knowledge entities in the domain ontology. The form-based interface of the competency sheets is configured by a competency ontology such as the one presented in the second section.

These editors define two structures, the first one related to actors' tasks and productions and the second one to the domain knowledge and competencies being processed by the actors. The next two tools, Competency Distributor and Evaluation Weighting serve to define associations between the two structures. Also, a Resource Manager can be used with knowledge and competencies to find and associate complementary resources not in the scenario.

These associations will be used to parameterize the evaluation tools. The Self Diagnosis tool can be used by any LKMA actor (learners, trainers, content experts, technical support, evaluators, etc.) to assess their own competencies linked to appropriate resources that they can use for support or learning. The Evaluation Grid is a tool for learners' evaluation by an assessment actor.

Finally, some of these evaluation results produced by the runtime actors can be integrated into an actor's model and/or into an e-portfolio, which is also called a learning and knowledge management product (LKMP). Competency evaluation results and the products of an actor's activities form the basis of their user model. The Actor's Model Integration tool serves to integrate and define the access to the LKMP data, according to a predefined policy that configures this tool. Then the actors' models can be exploited by a Generic Advisor tool that provides personalized assistance to the actors.

Figure 10 presents a global view of a competency management workflow in which these tools/services are used. It shows three phases. In the first phase, designers edit the LKMA basic components: a scenario workflow and a domain ontology (including entry and target competency). In the second phase, designers configure some evaluation tools that will be used by the scenario actors at delivery time. The last phase is when the evaluation of actual competencies is done and its results are integrated into actors' models and e-portfolios (LKMP). This is also where assistance data is added to the activity definition in the learning scenario.

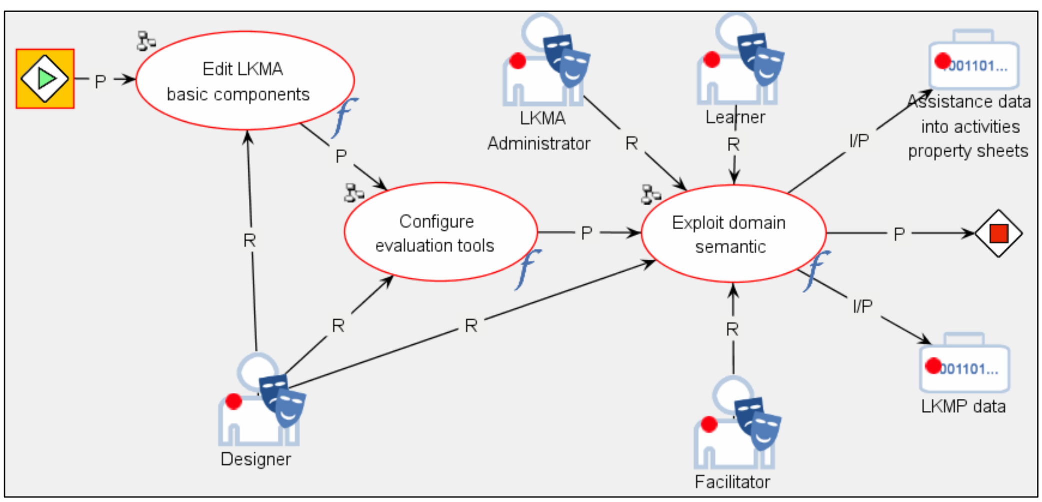

Figure 10. Global view of the Competency Management Workflow

Scenario and knowledge/competency design

Figure 11 describes in more detail the first phase. It shows the LKMA Designer's task using the scenario editor and the ontology editor with competency-property sheets. These edition activities produce on one hand a list of actors, activities, and resources present in the LKMA scenario, sorted by type, and on the other hand a list of competencies with their components. Both lists will serve to parameterize the competency distribution and evaluation weighting tools in the next phase. They can also serve, in the third phase, to design personalization rules based on competency gaps that will be triggered at runtime to provide to learners advice on resources to use, activities to complete, actors to consult, or simply on how to adapt their learning or knowledge management environment.

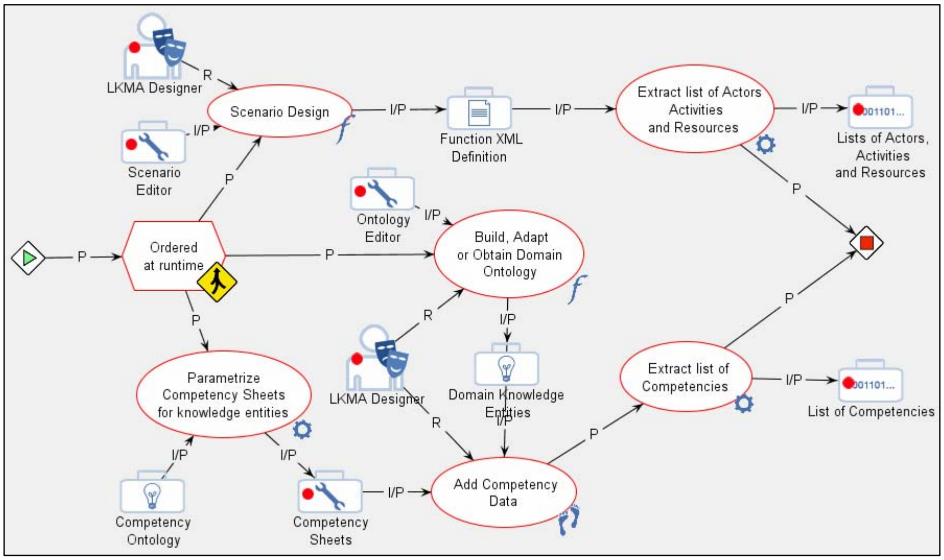

Figure 11. Design sub-processes for competency management

Knowledge and competency association

Figure 12 shows the second sub-process or phase of the competency management process.

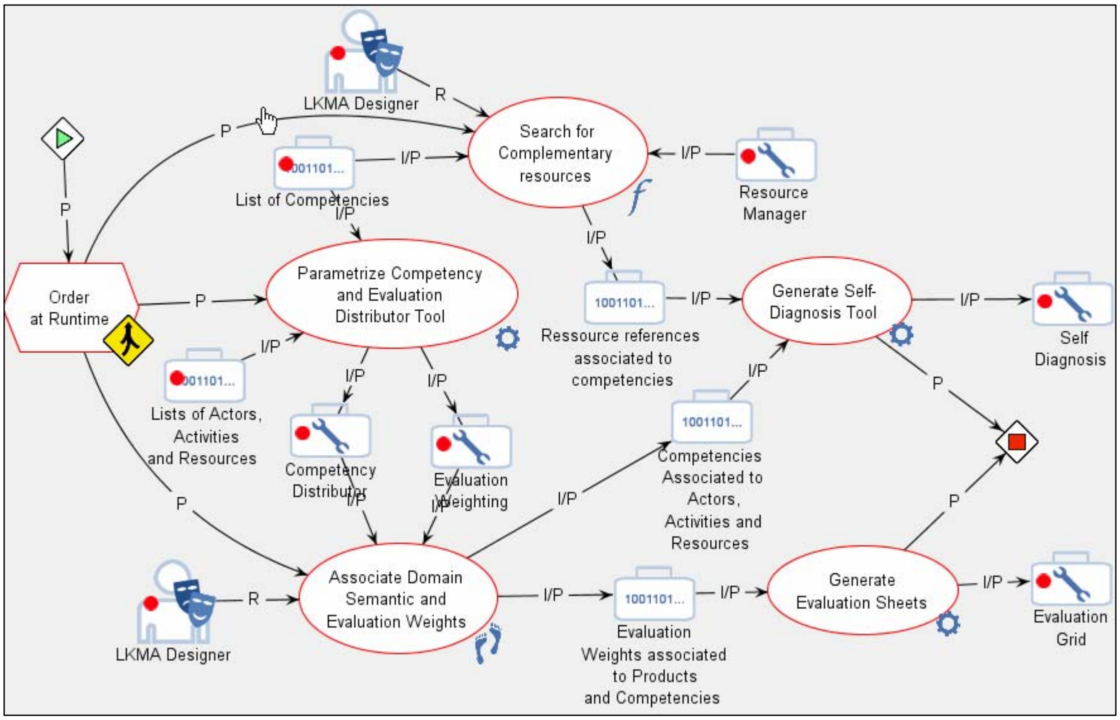

Figure 12. Knowledge and competency association sub-process

It is possible to annotate a scenario component directly by selecting it and opening a domain semantic interface showing the ontology editor and its competency sheets. But doing this in a piecemeal fashion is not the best way to design a coherent application.

To improve this, we need more synthetic association tools, such as the competency distributor and the evaluation weighting tools shown in figure 12. Basically, they take the form of two-dimensional tables with lines filled by a list of competencies already defined in the ontology editor. Column headings indicate the activities, the input resources, the products, or the actors in the scenario.

In the case of the evaluation weighting tool, only the learner products marked for evaluation by a designer are displayed in the columns. This tool enables the designer to distribute weights between them and, for each product, between target competencies for the product.

Competency evaluation sub-process

The above operations require a certain amount of effort by the application's designers that will be rewarded by the automatic production of corresponding tools for learners and facilitators to use at runtime for competency evaluation. The tools/services presented in figure 13 are called "competency evaluation tools" because they use the association of knowledge and competencies with actors, activities, and resources to provide evaluation and personalization functionalities.

The self-diagnosis tool used by learners (and possibly by facilitators), help these actors assess their own actual competencies with regard to target competencies defined by designers in the preceding phase of the process. The evaluation grid is used by facilitators to evaluate the actual competencies of learners.

Both tools output evaluation data that can be integrated into an actor's model and/or e-portfolio. This is done according to an administrative ontology (possibly including a digital rights management component) that defines how this data can be integrated into an actor's model and e-portfolio, and also what persons or computer agents will have access to it. The resulting LKMP data can be used to trigger personalization rules previously designed by a generic advisor. This assistance data will serve to control the flow of activities at runtime based on differences between actual competencies (stored in evaluation data) and target competencies set by the designers using the competency association tools.

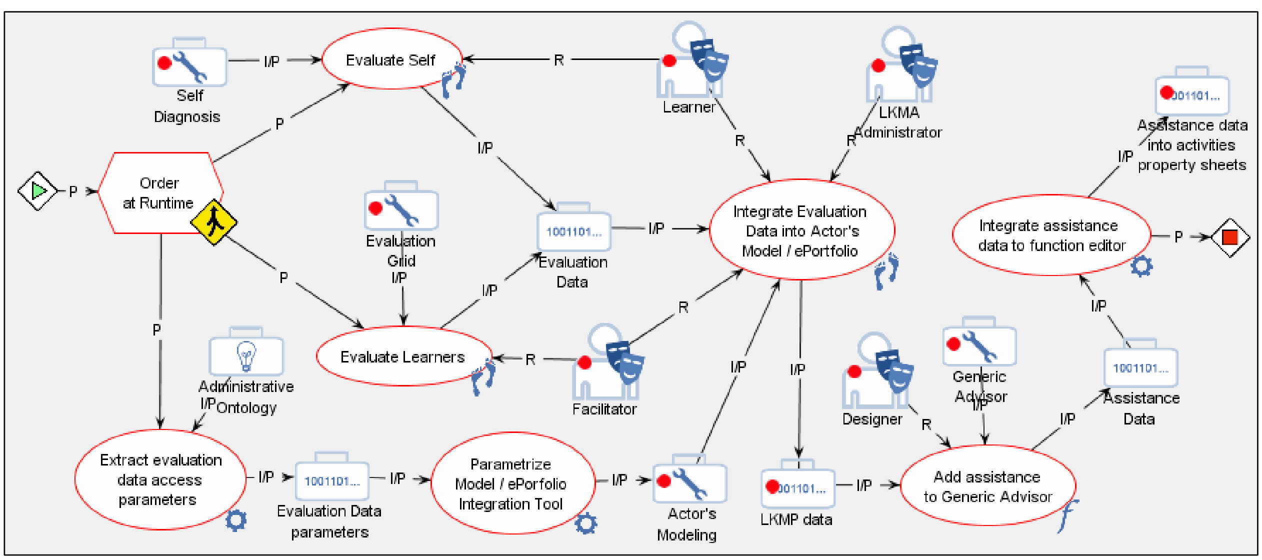

Figure 13. The competency evaluation sub-process

Conclusion

The importance given to competency management is well justified. Acquiring new competencies is the central goal of any education- or knowledge-management process. Thus, it must be embedded in any software framework as an instructional engineering tool, to inform the runtime environment of the knowledge that is processed by actors and their situation towards achieving competency-acquisition objectives. At runtime, the actual competency of actors must be evaluated in different ways, and this data can be integrated as a central piece for learners' e-portfolios and models. This data maybe scarce or elaborated, but it can be used by human facilitators or software agents to assist learners in acquiring new competency, and to adapt the learning environments to the learners' characteristics.

We are not too far away from releasing most of the tools presented in this framework. Most of them have already been built in the past, but the major challenge is to integrate them in a coherent, flexible, user-friendly, and scalable way, within the new context provided by the semantic web and the ontology-driven architecture of TELOS. What is yet to be proven is that the general approach presented here can be used at different level by average design practitioners and learners.

References

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American, May*, Retrieved June 7, 2007, from, http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21.

Bloom, B. S. (1975). Taxonomy of Educational Objectives: the Classification of Educational Goals, New York, NY: D. McKay.

Breuker, J., & Van de Velde, W. (1994). *CommonKads Library for Expertise Modelling*, Amsterdam: IOS Press.

Breuker, J., Muntjewerff, A., & Bredewej, B. (1999). Ontological modelling for designing educational systems. *Paper presented at the AI-ED 99 Workshop on Ontologies for Educational Systems*, July 18-23, 1999, Le Mans, France.

Chandrasekaran B. (1987) Towards a functional architecture for intelligence based on generic information processing tasks*. IJCAI-87 Proceedings* (pp. 1183–1192). Milan, Italy.

Davies, J., van Harmelen, F., & Fensel, D. (2002). *Towards the semantic web: Ontology-driven knowledge management*. New York, NY: John Wiley & Sons.

IMS-RDCEO (2002). *IMS Reusable Definition of Competency or Educational Objective Best Practices*. [Version 1.0, final specification]. IMS Global Learning Consortium, Inc. Retrieved June 7, 2007, from, http://www.imsglobal.org/competencies/index.html.

IMS-LD (2003). *IMS learning design: Information model, best practice and implementation guide*. [Binding document, schemas]. Retrieved June 7, 2007, from, http://www.imsglobal.org/learningdesign/index.cfm.

Krathwohl, D. R., Bloom, B. S., & Masia, B. B. (1964). *Taxonomy of educational objectives: The classification of educational goals*, New York, NY: Longman.

Martin, B. L., & Briggs L. (1986). *The affective and cognitive domains: Integration for instruction and research*, Eaglewood Cliffs, NJ: Educational Technology Publications.

McDermott, J. (1988). Preliminary steps towards a taxonomy of problem-solving methods. In S. Marcus (Ed.), *Automating knowledge acquisition for expert systems* (pp. 225–255). Boston, MA: Kluwer Academic Publishers.

Merrill, M. D. (1994). *Principles of instructional design*, Eaglewood Cliffs, NJ: Educational Technology Publications.

Paquette, G. (1999). Meta-knowledge representation for learning scenarios engineering. In S. Lajoie & M. Vivet (Eds.), *Proceedings of AI-Ed '99 in AI and Education: Open learning environments*, Amsterdam: IOS Press.

Paquette, G. (2002a). L'ingénierie pédagogique: Pour construire l'apprentissage en réseau. Montréal: Presses de l'Université du Québec.

Paquette, G. (2002b). Modélisation des connaissances et des compétences, un langage graphique pour concevoir et apprendre. Montréal: Presses de l'Université du Québec.

Paquette, G. (2002c). TeleLearning systems engineering: Towards a new ISD model. *Journal of Structural Learning, 14*, 1–35.

Paquette, G. (2003). *Instructional engineering for network-based learning*, Pfeiffer/Wiley Publishing Co.

Paquette, G., Crevier, F., & Aubin. C.(1994). ID knowledge in a course design workbench*. Educational Technology, 34*(9), 50–57.

Paquette G., & Marino, O. (2005). Learning objects, collaborative learning designs and knowledge representation. *Technology, Instruction, Cognition and Learning, 3*, 85–108.

Paquette, G., Marino, O., De la Teja, I., Léonard, M., & Lundgren-Cayrol, K., (2005). Delivery of learning design: The Explor@ System's case. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 311–326), Berlin: Springer Verlag.

Paquette, G., & Rosca, I. (2004). An ontology-based referencing of actors, operations and resources in eLearning systems. *SW-EL 2004 Workshop*, August 23, 2004, Eindhoven, The Netherlands

Pitrat, J. (1991). Métaconnaissance, avenir de l'Intelligence Artificielle, Paris: Hermès.

Polya, G. (1957). Comment poser et résoudre un problème? Paris: Dunod.

Popper, K. R. (1967). *The logic of scientific discovery*, New York, NY: Harper Torchbooks.

Reigeluth, C. (1983). Instructional theories in action: Lessons illustrating selected theories and models, Hillsdale, NJ: Lawrence Earlbaum.

Romiszowski, A. J. (1981). *Designing instructional systems*, New York, NY: Nichols Publishing.

Ruelland, D., Brisebois A., & Paquette, G. (2005). Supporting self-assessment in a competency approach to learning, *E-Learn 2005 Conference*, October 15-19, 2005, Vancouver, BC, Canada.

Schreiber, G., Wielinga, B., & Breuker, J. (1993) *KADS: A principled approach to knowledge-based system Development*, San Diego, CA: Academic Press.

Steels, L. (1990). Components of Expertise. *AI Magazine, 11*(2), 30-49.

Thayse, A. (1988). Approche logique de l'intelligence artificielle, Paris: Dunod.

W3C (2004). *OWL Overview Document*, Retrieved June 7, 2007, from, http://www.w3.org/TR/2004/REC-owl-features-20040210/.