Canadian Journal of Learning and Technology
Volume 28(3) Fall / automne, 2002

Organic Aggregation of Knowledge Object in Educational Systems

Gilbert Paquette

Ioan Rosca

About the Authors

Gilbert Paquette is Director at the Center for Inter-university Research on Telelearning Applications, CIRTA-LICEF, Télé-université, Université du Québec. Contact: Gilbert Paquette, 4750 avenue Henri Julien, bur. 100, Montréal, Québec.Telephone: (514)840-2747, ext. 2292; Email: gpaquett@licef.teluq.uquebec.ca

Ioan Rosca is a researcher at the Center for Inter-university Research on Telelearning Applications, CIRTA-LICEF, Télé-université, Université du Québec. Email: irosca@licef.teluq.uquebec.ca

Abstract

We propose an organic approach to educational web-based systems where learning objects, operations on these objects, and actors that perform them are aggregated in meaningful ways. The users of a learning system must be able to observe it globally, at different levels and from diverse viewpoints. They must be able to propose adaptations and improvements constantly using means of observation integrated with the means of action. For this, we need to provide inspectable and executable models of the learning system, to be used as prisms for understanding and control of operations. We propose to reference these models with educational ontologies developed for instructional engineering. The implementation of some of these ideas in the Explor@-II system provides examples. Conversely, the next Explor@ implementation will benefit from the discussion presented here.

Rsum

Nous proposons une approche organique pour les systèmes éducatifs distribués sur le Web par laquelle les objets d’apprentissage, les opérations sur ces objets et les acteurs qui les exécutent sont regroupés d’une intelligible. Les

utilisateurs d’un système d’apprentissage doivent être capables de l’observer globalement, à différents niveaux et selon des points de vue variés. Ils doivent être capables de proposer des adaptations et des améliorations en utilisant de façon continue des interfaces de visualisation intégrées avec des outils d’intervention. Pour cela, nous devons leur fournir des modèles inspectables et exécutables du système d’apprentissage, ces modèles étant utilisés comme prismes pour la compréhension et le contrôle des opérations. Nous proposons de référencer ces modèles sur la base d’ontologies éducatives développées pour l’ingénierie pédagogique. La mise en oeuvre de ces idées dans le système Explor@-II fournit des exemples. Inversement, la prochaine implantation du système profitera de la discussion présentée ici.

Introduction - An Organic Metaphor

Web-based distance education systems are complex. As mentioned by many authors (Harasim 1990; Paquette, 1995) these systems integrate many interrelated processes, aggregate a large set of components and can be considered from a wide range of viewpoints, paradigms and disciplines. A study of the theoretical bases of the Educational Technology (Rosca, 1999) shows the difficulties of building a rigorous yet operational synthesis.

We agree totally with those who seek to approach the learning object[1] paradigm from a pedagogical perspective (Koper 2001; Wiley 2002) rather than from a strict computer- engineering viewpoint. We believe that instructional engineering (Merrill 1994; Reigeluth,1983; Spector 1993) has much to offer on this question and our own R&D work on the MOT knowledge editor (Paquette 1996,1999), on the MISA instructional engineering method (Paquette, Aubin and Crevier, 1999), and on its ADISA web-based support system (Paquette, Rosca, De la Teja, Léonard, M., & Lundgren-Cayrol, 2001) has convinced us of both the complexity of the problem and of the need to develop an optimal strategy for the aggregation or learning objects.

With Wiley (2002), we disagree with the simplistic LEGO metaphor used by many specifications to represent the combination of learning objects. Wiley proposes an atom-molecule metaphor that suggests that learning objects cannot be assembled randomly with other learning objects because they have, according to their internal structure, a certain limited potentiality to combine with others. This metaphor is better but it is still insufficient. Learning object aggregation of two components does not result by mere reaction between them. We must consider not only the component objects, but also the context and the actors that produce the aggregation. Considering the anatomy of the aggregated system is not enough. We must also consider its physiology, its dynamics.

For this, we need an organic metaphor where cells are combined to form simple or complex organisms, where the whole is greater than the parts, produced by an operation performed by an external agent, here a designer or a user. The taxonomy of the educational resources cannot be understood apart from the taxonomy of the processes which create and process them. So we can speak of an ecology that would aim at the conservation of the " educational organisms " evolving by continual adaptation.

The purpose of this paper is to help a designer, including a learner acting as his own designer, to create a macro-organism from micro- organisms. How can such a designer observe, represent and plan a dynamic process, a metabolism, and at the same time, provide the underlying support and delivery structures?

Granularity and Types of Aggregation

At the heart of this general organic approach is the problem of knowledge object aggregation. This is an important educational goal. Small modular learning objects have nice software properties, but they tend to decompose knowledge into non-significant pieces. Furthermore, the information resources must be related to the operations and the persons enacting them. Learning objects need to be integrated into larger wholes that make sense to their users.

What are the desirable properties of interoperable learning objects? We observe that the integer-part problems that characterize the aggregation of learning objects is often evaded by those who want to make "factories" and "markets" of educational resources. The components must be sufficiently complex to be autonomous and useful, yet simple enough to easily integrate into larger educational sequences and activities. The ideal learning object behaves like a complete organism when alone and as a well-integrated being when aggregated as a part. It must have its own autonomy, the connectivity potential to combine physically or though communication, an encapsulation capability through an interface that concentrate its external relationships and some plasticity, that is the ability to adapt to the evolution of the aggregate.

To these intrinsic properties, we must add "economic" requirements such as the optimization of the engineering process and the reproducibility of the learning objects. A good component would be the one that allows its re-use most easily, most often and with the biggest gain of features. It is problematic to satisfy such contradictory conditions to arrive at the ideal model of content repurposing .

Key to the learning objects' repurposing issue, including multilingual support, is to develop adequate means for representing the objects at different levels of granularity and abstraction, and to support the sequencing, alternation and aggregation of learning objects into a meaningful instructional whole.

A stratification strategy, organizing the aggregation in hierarchical layers can help reduce this inherent complexity and establish a pyramid of meaning such as: {atom-molecules- compound, cell- tissue- organ- being- society},{letter- word- proposition- paragraph- text- literature} or {activity- lesson- course- program}. The analysis of aggregation strategies demands a sound conceptual basis, a specific terminology, the identification of primitives and some convention for the description of aggregation formulae.

We will discuss here different forms of aggregation we have built or prototyped while developing the Explor@-II system[2] . They represent a continuum of aggregation solutions. We do not pretend that this is an exhaustive typology, but we hope it will illustrate the organic approach to the aggregation phenomena. It should also put forward the advantages of the types of aggregation already implemented within the system and the need to develop new ones.

Fusion or Juxtaposition

In the first model of aggregation, several resources are combined to form a new educational object. Sections of documents are grouped multimedia documents. Sections of source code, executables and software components are combined to produce a new computer application. Components are integrated without modification (juxtaposition) or are modified (fusion) so that they participate in the aggregated object. For example, we can build a set of video sequences. In a juxtaposition mode, we just aggregate them in a web page and give access to the segments using hyperlinks. In a fusion mode, we add transition segments between the components to form a complete TV program. In such a situation, we must manage carefully the parallel modifications of the component source and of the aggregate during any revision or reengineering; if we change a segment, then the transition might become inappropriate and the aggregate will lose part of its meaning.

Composition Through Referencing

In another aggregation model, the components do not merge, but are deposited in a collection, in a grouping where group members share commonalities. We can use Learning Objects Models (LOM) specifications, such as Cancore (www.cancore.ca), to retrieve a set of objects having common characteristics. The search component of the Explor@-II resource manager will extract from a Cancore metadata referential all the resources corresponding to a set of constraints on their metadata attributes. This collection (the record set) is another type of aggregate, composed on the basis of referencing similarities.

Control and Filtering

In this third model, the aggregate is composed of a “master” object A that controls one or several “slave” objects B. These slave objects do not lose their integrity but are “covered” by A, which acts as a filter interposed between the user and the slaves, controlling access to the primary B objects. For example, the resource controller of Explor@-II is such a master object. The resulting aggregate of A plus the set of Bs can be called either “a controlled set of primary resources” (the Bs) or “a secondary resource” (see Figure 3). In a simple case, the controller only enacts a necessary preparation to the launching of the B objects, for example, to prepare a text file resident on a server for its opening in a text editor on the user’s station. The role of the controller can of course be greater than that. It can intercept and remember actions of the components, transmit data to other components, insert commands to other resources, offer prefabricated or dynamically built pieces of advice as in intelligent tutoring systems or advisors. The Explor@-II advisor system is such a control aggregate. We can design controller objects for filtering particular operations and tools such as an email or an Internet browser or we can design generic controllers, capable of taking care (aggregating) a wide range of resources. Examples of large and generic controllers are expert system “shells”, operating systems, LMS and LCMS. The Explor@ system itself, as well as its main modules, in their controller aspects, are generic aggregators of resources.

Scripting

The fourth model of aggregation is the workflow planning performed by a human or computer agent, sometimes called scripts. A designer plans a series of tasks by means of an editor, describing the organization of the operations in a text, hypertext or graph format. The resulting document plan aggregated with the operations it contains, provides a sequencing and branching structure. Rather than simply executing the script and launching the corresponding resources (operations or documents), we can do it through an object controller, thus combining the present model with the previous one, then passing from secondary to tertiary objects. For example in the Explor@ system, a designer has a tool allowing him to describe a scenario of learning activities and to bind the activities to resources needed to perform them. In the corresponding learner environment, the user can launch these resources, but also answer questions, receive advice, etc.

Coordination (Emergent Aggregation)

In all the previous aggregation modes, there is only one actor using the aggregate composed of operations, tools and documents, but in a distributed learning system we need larger aggregates where different actors interacts with one another through operations, tools and documents. Furthermore, the planning of such an educational aggregation is not always possible or desirable: the persons participating in a learning event in fact define an integration process or metabolism in an emergent way.

Here, it is the communication (cooperation) that structures the aggregate. We call this process coordination. There is no fusion of entities or wrapping objects. The integrative layer operates at the process level. The only materialization of the composition is the discourse found in the communication channels. This communicational connectivity is supported by structural links, by innate and educated communicative activity.

This type of aggregate is what we call a “function model”. Such an inspectable and executable model aggregates resources used or produced by users, with operations that these users perform and possibly other functionalities linked to an operation such as a data entry form, a communication link or some assistance services. A learning activity scenario, a competence evaluation or a resource management process, are all examples of a function model. These models can provide dynamic user interfaces where a person can use resources to perform an operation and produce new resources for him/herself or others. A function model can group operations according to a typical actor’s role such as a learner consulting information or a trainer doing evaluation or coaching. The function model aggregate can also group operations for multi-actor coordination such as an author building a test, a learner passing the test, a trainer evaluating the test, the learner again reorganizing his/her study plan (see Figure 1).

Let us come back for a moment to the three metaphors presented at the beginning of this article, the Lego, chemistry and organic metaphors, to see how they can account for the various types of aggregation we have just presented.

Table 1 compares the piece/assembly/motor, the atom/molecule/reaction/manipulation and the cell-tissue-organ-organism analogies. We see that the Lego and chemistry metaphors are as good as the organic metaphor to represent three types of aggregation: juxtaposition, record set and control/filtering. They can also represent to a certain extent scripting aggregations if we add the idea of a plan or program, but such a plan is not really a part of the aggregate, while the biological processes aggregating organs are built in the organism. The first two metaphors are even more powerless to account for coordination aggregates.

Table 1. Aggregation types: A comparison of the Lego, chemistry and organic metaphors.

AGGREGATION TYPE	Lego Metaphor	Chemistry Metaphor	Organic Metaphor
Juxtaposition /Fusion	Assembled objects from blocks or components	Molecules from	Tissues from cells
Record Set	Similar pieces	Similar molecules	Similar tissues
Controlled set of objects	Motors (as Lego pieces) acting on other pieces	Chemical reactions with a catalysis	Glands controlling the reactions of other organs
Scripts	Program controlling motor operations	A suite of steps in a labexperiments	A set of organs reacting in a biological process
Function Models (Coordination aggregates)	No communicating agents or actors	No communicating agents or actors	An organism as a set of organs coordinated through communication

The organic metaphor offers a useful analogy for coordinated aggregates, that of a society of organs interacting with the others. For example, let's look at the coordination of organs in an animal's body. An animal (seen as a coordinated aggregate of organs) sees another animal moving, the vision system enacts a set of operations that send images to the brain, the brain system interprets this information through a certain number of operations, comparing it with some patterns in its memory, the brain sends a command to the motor organs, the animal starts to run.

Inspectable and Executable Function Models

A fundamental question arises: who can create and maintain a macro-educational resource? An author using editing tools can design it but it can also result dynamically within a community provided with communication, negotiation and coordination tools. Can we also implement the essential operations of aggregation in a mechanical or automatic way, such as proposed sometimes by the CBT industry? For the moment, while observing with interest the efforts in artificial intelligence and also using them to a certain extent, we are cautious because wrapping components in a global resource must be based on a purposeful pedagogy that cannot be deduced automatically from some predefined specification, except in very simple cases. The approach that we propose is a symbiosis between the complementary capacities of the human and artificial agents to facilitate the aggregation of the resources.

Based, on this approach, functions models, or if you like, coordination aggregates, represented by the dynamic nature of an organism's physiologies, help describe the complex coordinated operations in distributed learning environments, where human actors are aggregated with the operations they perform, and the resources, documents and tools they use, or produce for other actors and their corresponding operations. Here, the aggregated process or workflow is in fact a use case of the computerized learning support system.

Coordination Models and Reality

This solution entails that the descriptions of the aggregation formulas for the objects and processes are readable by human beings (semiotic and ergonomic criteria), but also by computer programs (technical criteria). The search for a "common denominator" between these descriptions brought us to the MOT representation system (see Figure 1) that offers a dual readability. The graphs representing the internal structure of the aggregation (nodes and links) can be translated as XML or relational data to be interpreted by computer agents monitoring the operations and the progress. The graphic representation of the same structure, enriched by textual comments, offers an interface to the human actor. We arrive here at a key idea: the use of the model as a support for the actual aggregation. Let us observe the pair formed by a real coordination learning process and its description in a function model. The real constituents are reflected by their images in the model, the aggregate being represented by the global model. If we realize a good correspondence between the reality and the model, an inspired morphism, it is possible that certain phenomena are easier to observe from the point of view of the image than watching in the reality. For example, a graph such as on Figure 1, representing a process composed of several integrated procedures, their inputs and products, facilitates the observation of the process, which, because of its sequencing in time, is difficult to analyze in a direct way. If the model is analyzed after the fact, it is a description of what took place; the gain here can be a better understanding of the past experience, which can influence future actions. If the model is used before the action as a plan or specification, the gain is the capacity to orient the real operations of the actors in a favorable direction. If the model is built during the action, in a emergent way, we find, particularly for a project-based learning situation, the dynamism necessary for the communities of learning to build and develop.

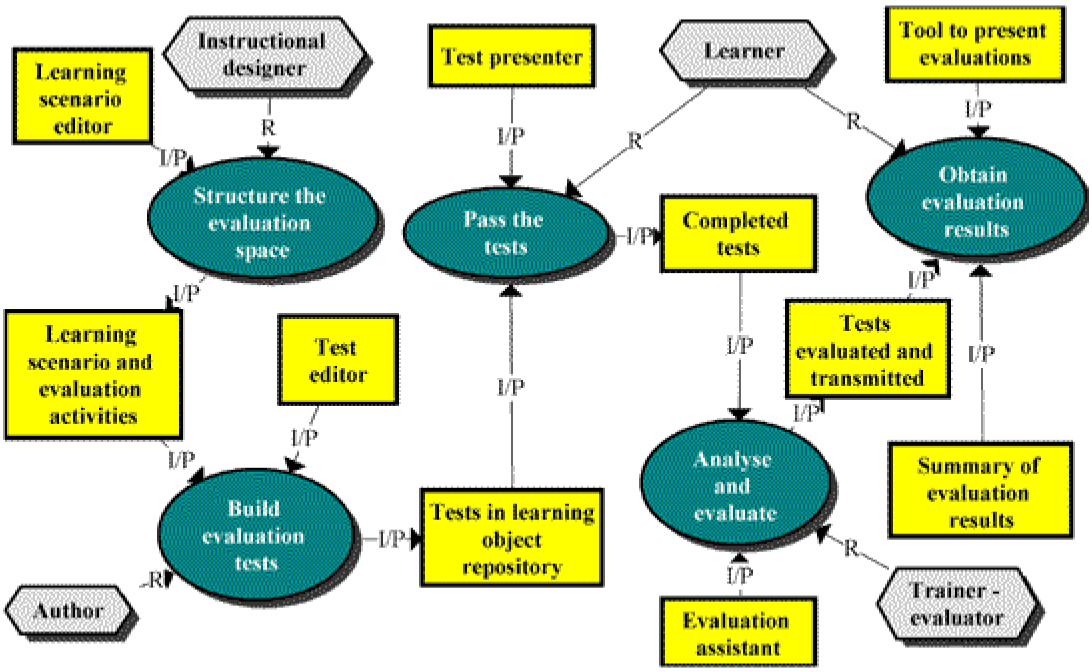

Figure 1.

An inspectable and executable function model for the evaluation process.

Generally speaking, the executable function model is used as a communication and control interface where a cause-effect loop appears: an actor can change his behavior towards an operation or a resource, because the model allows him to perceive the context of his/her actions. A global aggregation between the model and the reality takes place.

An Aggregation Facilitator

If the actors can communicate, launch resources, start operations by acting on the model acting as a user interface, the model becomes the aggregate, which binds and defines the context of use of the component resource. A page that describes a series of documents reflects a collection. But if we launch documents from it, it becomes a layer of actual aggregation. An executable MOT graph, such as the one on Figure 1, giving access to the resources and the communication services, can achieve an aggregation of the structures and the processes. A distributed delivery model (Paquette and Rosca, 2002) is such an aggregation, an editable mini

LMS or LCMS (depending on the nature of the operations), which can in his turn be aggregated with other LMS or LCMS.

The role of a system like Explor@ is to facilitate the aggregation of objects, operations and actors by using the various formulas of aggregation previously presented.

As illustrated in Figure 2, the use of function models as an interface for aggregation is not always necessary or useful. Other facilitating tools such as the resources controller, the actor's roles manager and the objects manager and it metadata search engine are also parts of our aggregation arsenal. The purpose of all these tools is to help the actors to track down the components, to negotiate their use, to put them in function in an adequate technical and pedagogical context.

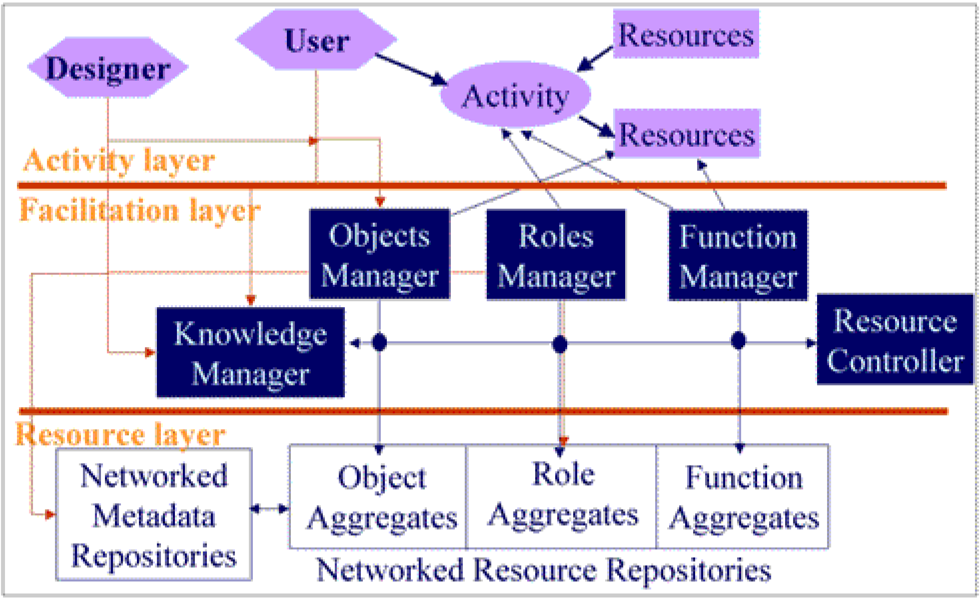

Figure 2

. The main modules of the Explor@ facilitation layer.

The aggregation facilitation layer, which is the core of the Explor@ system, is situated between the user's interfaces providing access to the single or aggregated objects, sometimes grouped according to their roles as an actor or their interactions in a function model, and the networked resource repositories available through the front end metadata repositories. The repositories contain the simple and the aggregated resources such as compound objects, roles and functions, that are built and reintroduced as new resources into a repository.

Abstracting Resources

A particular effort has been made to support designers of distributed learning environments. The various design of facilitators, actors, content experts, trainers, technicians and managers, use different languages and intervene in different life phases of the learning objects. Depending on the actors, the same object can be named information, lesson, file or document. It is worth looking for a language and for interfaces specialized in supporting the interventions of different actors' types, while assuring a mutual translation to preserve the general coherence.

From this originated the idea to wrap the individual resources, called primary objects, into a secondary layer of control objects, called secondary objects. This layer provides a unitary interpretation by the Explor@ virtual machine, acting as translation services offering a terminology for triggering the primary object's functionalities, in a language appropriate for different categories of users. The primary and secondary objects can then be wrapped, grouped and sequenced in tertiary objects by the designers.

A technician will concentrate on the programming of the primary objects to implement the features defined in the secondary control object, thus offering a set of scripting primitive. The education specialist will compose an aggregation starting from abstract secondary resources, scripting the tertiary object features in the language of its choice. These operations enable the designers to perform most of the aggregation processes described in Section 2.

The end user will use function maps allowing him to manipulate primary, secondary and tertiary resources, which were associated to nodes. Figure 3 illustrates the three levels of objects and also their possible aggregation within a function model.

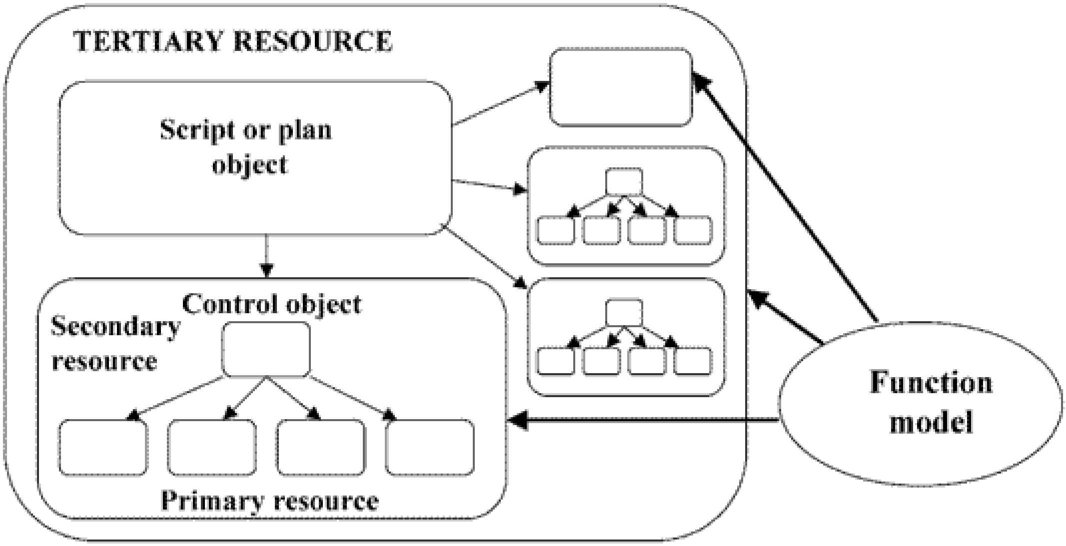

Figure 3. Encapsulation of primary objects into secondary and tertiary objects.

The Function Model Interface

We now present a mock-up of a function model user interface. Several prototypes have been built to explore the possibilities. The intent here is to replace the traditional single-user tree structure (course, modules, units, activities, documents) found in most LCMS on the market, by a network of actors, operations and resources, giving the users a graphical view of their interactions, not only within operations or activities, but also with other users playing different roles in a process. Figure 4 shows such a user interface. The function model on Figure 1, representing a learner's evaluation process, has been embedded here in a browser. Let us focus on the central operation "Pass the test". If any user clicks on the corresponding oval box, it will open a resource displaying the assignment that helps perform the operation. Now the actor performing that operation, a "learner", can click on the hexagonal box, opening an aggregated resource, here a role interface giving him/her access to a set of resources according to his roles grouped as self-management, information, production, collaboration or assistance. Then, if he/she clicks on the rectangular box, labeled "Tests in learning object repository", he/she will access a resource put in there and revised from time to time by a test author. On the figure we have shown a kind of test different from a traditional Q&A test, to show the versatility of the learning object repository paradigm. Here the test is a digitized video resource. Of course the assignment would explain what to do with the video; for example, the learner would have to describe in his/her own words the operations performed by the technician on the video.

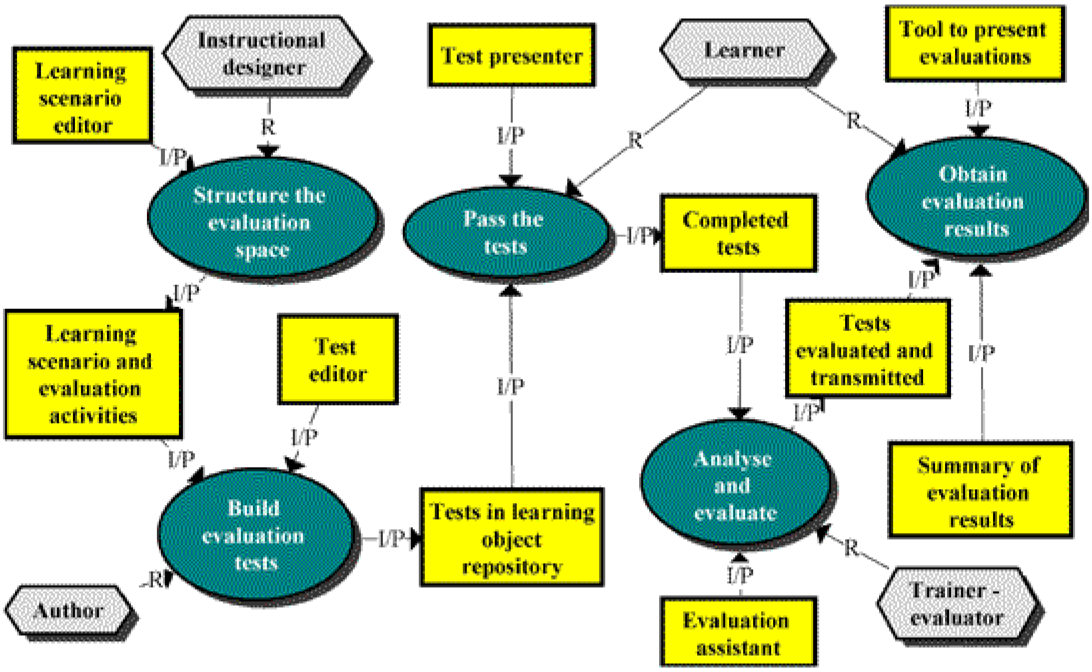

Figure 4. A function model user interface.

Knowledge Referencing of Objects, Operations and Actors

Within the resource manager, part of the metadata description of a learning object can characterize their content by pointing towards terms in a domain ontology. This scheme can be used to reference in a common framework the actors, the operations and the objects within a function, as well as the function model itself, as a resource aggregate. It is natural to think of a common reference ontology because the resources and the activities are linked as elements of the function map. This observation is valid also for the description of the actors: the indexing of their competence can use the same reference ontology as the resources and the activities, as shown on Figure 5.

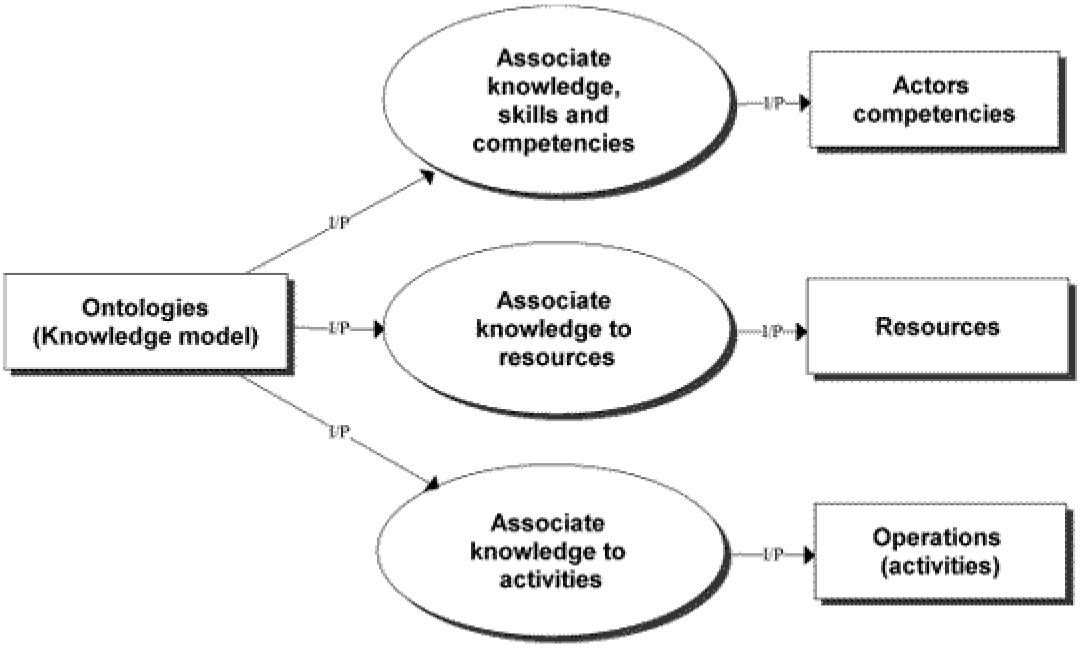

Figure 5. Knowledge referencing of resources, operations and actors.

The use of a unique knowledge reference facilitates the communication between the actors (as user representatives), the object resources and the activities or operations within the framework of a function. It allows the display of "mixed record sets" which gather objects, activities and relevant experts for the learning of a certain subject. For example, a user could query the system for a certain subject and obtain three learning objects to consult, two experts to communicate with and four activity modules or operations to perform, all pertinent to the subject. This unified knowledge reference also allows a global regulating reaction: if the user of a resource indicates gaps in the ontology describing the knowledge related to it, his observations can propagate to the indexing of some activities and of some expert competency to fill other gaps in the system.

By observing the connections created by a common semantic reference to the objects, operations and content experts, we uncover a form of aggregation that illustrates the subtlety of the educational metabolism. It also reveals the aggregating nature of the language and the interfaces shared by a community of practice. It invites us to consider this type of aggregation in the framework of the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001).

Such a referencing of the different resources of a learning system by a knowledge model, was first built as early as 1992 in the MISA learning systems engineering method (Paquette, Crevier, and Aubin, 1994) and recently implemented in the educational engineering system called ADISA (Paquette et al., 2001). ADISA supports a designer in the explicit association of knowledge sub-models to learning activities or operations and learning media instruments, while defining the actual and targeted learners' competencies in a learning system.

Towards a Set of Interoperable Lms'S

The Explor@ virtual machine managers can be used independently. Each is also decomposable into modules that can be integrated or not into a particular application. The purpose here is to create a set of editable, combinable and interoperable mini-LMS rather than to add another big LMS to those already flooding the market. In fact, each function model or scenario is a specialized learning management system (LMS). We can group these LMSs into larger wholes to form a more or less complete LMS, seeing them also as learning objects that can be aggregated. This nice plastic property makes possible for a system like Explor@ to flexibly integrate part of its operating components with other LMSs, LCMSs or distance learning platforms.

Explor@ integrates the concepts of "Learning Objects Repository", "Learning Management system" and "Learning service provider". It can be used as a autonomous system, to satisfy the distributed clients of a virtual campus or a community of practice. The architecture emphasizes the dynamic and flexible integration of various forms of aggregation. Because of its modularity, Explor@ can be also be used as a support system for an external LMS which needs its services. The combination between Explor@ and the LMS which it strengthens can be achieved by merging the two architectures or by communicational links, provided that a protocol of data exchanges and methods declaration is established and respected. To facilitate this interoperability, Explor@ will use XML exportable data formats, will offer interfaces for the external calls of its services, will provide an abstract functionality layer.

The architecture of Explor@ will thus be based on an internal "services BUS" which links clients and suppliers modules. This will allow Explor@ to fit naturally into a network of learning objects directory and aggregation tools. It will offer to the external network some macro-resources composition and management services, using the other services suppliers accessible on the network.

This is a consequence of the organic approach proposed here. The decision to make Explor@ a cell in a vast network generates an unavoidable concern for standards allowing the communication with the other parts of the organism that will contain it. Without losing its capacity to work alone EXPLOR@ will become interoperable in a network of Learning object repositories by using resources description and interoperability standards such as IMS, CANCORE, SCORM, AICC, etc. For the interoperability with the Learning Management Systems, Explor@ will sustain the efforts to normalize the descriptions of the aggregations formulae and of the macro-resources metabolism.

A promising alternative to static aggregation is the Web services approach (Kreger, 2001) that integrates objects through communication methods. This dynamic form of integration leaves the objects in their actual locations, while making use of some or all of their services whenever needed by other objects. This approach enables the objects to change dynamically their linkages while turning around the practical difficulties that arise when one starts using an object outside its original context.

In line with the new heavy trends encouraged by major players like IBM, SUN and Microsoft, Web services use HTTP and XML standards as an exchange language between objects, the SOAP protocol for requests to services, WSDL for the declaration of services and the UDDI registration scheme to track down services on the network.

We consider that the new trends in the world of Web infrastructures, beyond the technical details, stems from a new sensitivity to the organic character of the systems. We seek more and more a soft and dynamic aggregation of objects. The remote communication between components eliminates the need of morphological fusion that raises impressive technical problems.

Conclusion

If learning objects are to be single-purposed, of use only in a single context, and only appropriate to a single level of granularity and abstraction, then the value of learning object repositories will be seriously impaired. The learning object is a raw material that can be used in different ways. It is the activities you do with it and their integration in meaningful scenarios or functions that count. For this, we need a very flexible educational operations system that goes beyond fixed distance education platforms and LMSs, and that can complement other platforms or LMS by providing new repurposing capabilities. The instructional operation system

Explor@ allows this type of approach because of its flexibility.

Analyzing a recent review of distance learning platforms (Harmbrech, 2001), we have noticed that the current platforms are designed for predefined actors, usually providing a fixed set of tools and resources for an author, a learner, and sometimes, a trainer. On the contrary, the open and versatile framework presented here allows for any set of actors without predefining the functions or roles. It allows the implementation of interactions between actors using resources dynamically related to the operations the actors perform in the system. Hence, within the same system, by aggregating objects, roles and functions, it is possible to build very different distributed learning systems such as electronic performance support systems (EPSS) in a workplace, communities of practice or, at the other end of the continuum, formal distributed classroom activities.

A coherent management system of a network of educational resources / services directories demands adequate solutions for the problem of the composition of macro-resources / services. The integration of the structures or the processes can be made by the sharing of the reference ontologies, by the coordination of the communication between the actors, by interoperable technical infrastructures, by standards of representation of the aggregation formulas, by use of aggregation facilitating interfaces. The future Explor@ system implementation will consider all these strategies as ways to implement an organic vision on the educational phenomena, through graphic models used as facilitating interfaces to interact with dynamic aggregated resources.

References

Berners-Lee, T., Hendler J., & Lassila.O. (2001, May). The Semantic Web . Scientific American.

Harasim, L. (1990) (Ed.). Online education: Perspectives on a new environment . New York : Praeger Publishers.

Harmbrech, W.R. (2001). Corporate e-learning: Exploring new frontiers .WR Harmbrech & Co. Retrieved May, 15, 2002 from elearning@wrhambrecht.com.

Koper R. (2001). Modeling units of study from a pedagogical perspective The -pedagogical metamodel behind EML. Retrieved May, 15, 2002 from http://www.eml.ou.nl/introduction/articles.htm.

Kreger H. (2001). Web services conceptual architecture (WSCA 1.0) . IBM Software Group.

Merrill M.D. (1994). Principles of Instructional Design. Englewood Cliffs, New Jersey: Educational Technology Publications.

Paquette, G. (1995). Modeling the virtual campus. In B. Collis and G. Davies (Eds.). Innovating Adult Learning with Innovative Technologies. Amsterdam, The Netherlands: Elsevier Science B.V.

Paquette G. (1996, avril). La modélisation par objets typés: Une méthode de représentation pour les systèmes d’apprentissage et d’aide a la tâche. Sciences et techniques éducatives , 9-42.

Paquette, G. (1999, June). Meta-knowledge representation for learning scenarios engineering. In S. Lajoie et M. Vivet (Eds.).

Proceedings of AI-Ed’99 AI and Education, open learning environments

. Amsterdam: IOS Press.

Paquette G. (2001).TeleLearning systems engineering: Towards a new ISD model. Journal of Structural Learning, 14, 1-35.

Paquette G., Aubin C., & Crevier, F. (1999, August). MISA: A knowledge-based method for the engineering of learning systems. Journal of Courseware Engineering , 2.

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

Paquette G. & Rosca I. (2002). Modeling the delivery physiology of distributed learning systems. Manuscript submitted for publication to Technology, Instruction, Cognition and Learning.

Paquette,G., Rosca, I., De la Teja, I., Léonard, M., & Lundgren-Cayrol, K. (2001, October). Web-based Support for the Instructional Engineering of E-learning Systems. Paper presented at the annual WebNet Conference, Orlando, FL.

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

Spector J.M., Polson M.C., & Muraida, D.J. (Eds.). (1993). Automating Instructional Design, Concepts and Issues. Englewood Cliffs, New Jersey: Educational Technology Publications.

Rosca, I. (1999). Vers une vision systémique du processus de l'explication- Récit d'une recherche sur l'intégration de la pédagogie, de l'ingénierie et de la modélisation . Thèse de doctorat non-publiée, Université de Montréal, Québec, Canada. (http://www.iro.umontreal.ca/~rosca/index.htm).

Wiley D.A. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D.A.Wiley (Ed.). The Instructional Use of Learning Objects. Bloomington, Indiana: Agency for Instructional Technology and Association for Educational Communications of Technology.

Endnotes

1. In this article, we will use the terms learning objects, knowledge objects and resource as synonyms, but we prefer the term resource, or more precisely educational resource because it embeds resources intended for designers, managers, trainers as well as learners, and also because the resource can be a knowledge object (embedding content) as well as a tool or an agent representing actors or categories of persons, and the operation or function they perform.

2. Explor@-II is a new version of a distance learning support system, under development since 1992. In 1995, an object oriented model of a virtual campus, of its actors, activities and resources, has been built with a prototype platform. In 1999, an operational system has been put in operation at Télé-université where it support hundred of students. The Explor@-II will replace the former system in the fall of 2002 at Télé-université and to support the delivery services at Canal Savoir, a consortium of Quebec universities and colleges. New developments are starting within the eduSource project on learning repositories funded by CANARIE.

ISSN: 1499-6685