

# Modelling and delivering distributed learning environments

Gilbert Paquette

*Center for Inter-university Research on Telelearning Applications, CIRTA-LICEF, Télé-université, Université du Québec, Canada*

**Abstract:** Delivery models are important because they represent the actors, their operations and interactions, and the resources represent the use or product for other actors when the system operates in the future. Without sufficient planning, distributed learning environments will generally present high levels of technical and organisational noise that are an obstacle to learning. We will present a delivery model technique which aims to solve these problems, especially in the context of flexible life-long learning and distributed professional development systems.

**Key words:** delivery models, instructional design, use of case modelling, distributed learning, professional development environments, web-based learning, learning object aggregation

## 1. INTRODUCTION: THE DELIVERY CHALLENGE

A delivery model depicts the interactions between users and learning system components (hence its physiology), in order to plan (before), to facilitate (during) and to describe (after) the processes represented. Such a model represents the actors and their interactions with the resources they use or provide to other actors while performing operations. These resources have to be in place when learners start using a learning system (see Paquette et al., 1999). Delivery modelling also allows the designers to plan for management personnel with the financial, organisational and logistical information required to create and maintain the learning system throughout its useful life.

Along with the learning and teaching strategy selected through instructional design, delivery modelling and planning is certainly the most

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35615-0\\_52](https://doi.org/10.1007/978-0-387-35615-0_52)

determining factor in terms of the success or failure of a distributed learning environment (DLE). One could argue that delivery is the most important modelling area for educational technologists because it describes the real time use of the learning system, instead of the initial somewhat theoretical view of that system by a subject matter expert or an instructional designer. The delivery model is the sole provider of a global and synthetic representation of a learning system needed to manage complex and hybrid phenomenon units that involve individuals, objects, information and concepts amalgamated for learning purposes.

Generally speaking, much importance is given to the media involved in building the materials of a learning system. However, in many distributed learning systems, few new materials are produced. Regardless of the type of delivery model, the selection of resources, not only materials, is always necessary to build a useful distributed learning environment. In DLE models such as hypermedia self-paced learning, on-line learning, learning communities of practice or electronic performance support systems (EPSS), disfunctional resources can amplify difficulties and create a 'technological noise' that prevents learning and teaching. Organisational noise also results from poor co-ordination between the staff that support training when some people lack knowledge and tools regarding their tasks in a given learning event. Finally, a 'comprehension noise phenomena' emerges when the planning facilities do not provide interaction observation facilities on the spot.

This article presents a delivery model technique which aims to solve the aforementioned problems.

## **2. BUILDING DELIVERY MODELS**

Creating and documenting one or many delivery models are the most critical delivery planning tasks. An example of a model created by the author and currently used in an artificial intelligence course offered by Télé-université du Québec, highlights the interaction between six types of actors: learners, instructors (also called tutors), designers, managers, network administrator, and shipping clerks. These actors perform various operations: using the distributed learning system (DLS) that includes web content and printed material, ensuring pedagogical support, creating and maintaining the Explor@ website and the networks, and publishing and updating the course website mailing material.

The model offered presents many types of resources: three types of material: the course website, the Explor@ website that manages the actors' environment and the non-computerised resources (books, videotapes); two

means of communication: the internet and regular mail; many tools: software packages, web browser, TV station and VCR to view videotapes at home, an HTML editor and other media authoring tools to update the website, as well as servers and software to create and maintain the Explor@ environment and the network components; services offered to the participants, as well as the locations where activity takes place: the participants' home where all learning activities are undertaken and a warehouse from where the course material is shipped. This example concretises the concept of delivery models.

### **3. FROM FUNCTION MODELS TO ACTORS' ENVIRONMENTS**

We will now take these concepts a step further and describe how various delivery models can help create distributed learning environments that provide the necessary resources to the actors involved. This procedure is composed of three main phases: a) building models that represent the functions within a learning system, b) identification of the actors' operations and resources within a function, and finally, c) the creation of the corresponding actors' environments integrating learning materials, tools and other resources.

#### **3.1 Functions or use cases as delivery models**

*A function within a learning system* is a type of delivery model that corresponds, from a computer science point of view, to a use case of that system. Knowledge management, competency management, learner evaluation or resource management processes are all examples of a function. From a conceptual point of view, in a biological or ecological metaphor, a function is a particular physiology, an interesting subsystem of operations within the learning system organism.

Technically, a function model is a sub-model of the delivery model considered in Section 2. Designers first structure the evaluation space using instructional engineering tools to define scenarios, activities and evaluation materials. Then, authors create tests that are referenced with the other learning objects and made available to the other actors' through a learning objects repository. A learner finds an appropriate test, completes it and sends it to a trainer for evaluation. Trainers compile marks and annotations that are placed in an assessment bank, available to the learner and the training

manager. The learner obtains his/her evaluation results and makes decisions using this information.

### 3.2 Identifying resources for the actors' environments

Functions provide an inventory of the resources used or produced by different actors. Table 1 compiles these resources, tools (T) and learning materials (M), for four actors (learner, designer, content expert and trainer) involved in three functions: learners' evaluation, competency management, and knowledge management.

Selected Actor's Name	Function	Material and Resources Used	Material and Resources Produced
LEARNER	Learners' evaluation	(T) Test presentation tool (M) Test in learning objects repository (T) Tool to present evaluation results	(M) Completed test
	Competency Management	(T) Test presentation tool (M) Competency dictionary and profile	(M) List of target competency
	Knowledge Management	(M) Model references in repository (M) Existing knowledge model	(M) New model
DESIGNER	Learners' evaluation	(T) Learning Scenario Editor	(M) Scenarios and evaluation activities
	Competency Management	(T) Competencies Editor	(M) Competency dictionary and profile
EXPERT	Learners' evaluation	(T) Test style sheets (M) Scenarios and evaluation activities	(M) Tests in a learning objects repository
	Competency Management	(T) Test presenter	(M) Competency Offer
	Knowledge Management	(T) Model Editor (M) Model Library (M) Model structures (M) Knowledge base	(M) New Knowledge Model (M) Presentations, Materials, etc (M) Model referenced in repository
TRAINER	Learners' evaluation	(T) Evaluation Assistant (M) Completed test	(M) Evaluated Test
	Competency Management	(M) Collaboration Assistant (T) Registering Tools (M) Learning events repository (T) Management and Exportation Tools	(M) Competency registry (M) Advice on learning events and experts (M) Transfers to Competencies registry

Table 1. Resources by actor and by selected function

Indirectly, these resources specify the actors' operations or roles. As displayed in Table 1, in the function 'Learners' evaluation', the designer produces assessment activities. The content expert will consult these descriptions and use a test editor to generate tests, which are stored in a

learning object repository. In the 'Competency Management' function, the expert takes a test to assess his/her competency that will be displayed as 'competency offer' or mentorship possibilities to the learners.

## **4. AGGREGATING RESOURCES**

Using this list of resources used or produced within each function, we can create an environment for each actor. For this we use the delivery system Explor@ (Paquette, 2001).

### **4.1 Building the actors' environments**

Explor@ designers' tools help define three groups of resources (called spaces) for the expert corresponding to the three functions of Table 1. The designer selects or adds resources to the learning object repository if necessary using an IMS-CANCORE metadata-enabled resource manager. Then, he/she specifies the interactive spaces associated to each function and the corresponding resources that the expert needs to perform his/her operations in each of the functions.

The resulting expert's environment window adds to the course web site three menus that offer the resources identified for this actor. Other environments can be created from Table 1 for the other actors as well.

### **4.2 Learning object aggregation**

Aggregation of resources is an important educational goal. Small modular learning objects have useful software properties, but they tend to decompose knowledge into non-significant pieces in a Taylor-like fashion. 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.

We agree totally with those who seek to approach learning object referencing and aggregation problems from a pedagogical perspective (Koper, 2001; Wiley, 2002). With Wiley, we disagree with the simplistic LEGO metaphor used by many to represent the combination of learning objects. We believe that instructional engineering (Reigeluth, 1993; Spector, 1993; Merrill, 1994) has much to offer on this question and our own research and development work on the MOT knowledge editor (Paquette, 1996; 1999), on the MISA instructional engineering method (Paquette et al., 1999), and on its ADISA web-based support system (Paquette, Rosca et al., 2001)

has convinced us of the complexity of the problem and of the need to develop an optimal strategy for the aggregation of learning objects.

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 into learning 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 Learning Management Systems (LMS) by providing new content repurposing capabilities.

Analysing 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. The instructional operation system Explor@ provides an open and versatile alternative. It allows building environments 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 resources and functions, it is possible to build quite different distributed learning systems such as electronic performance support systems (EPSS) integrated in a workplace activity, communities of practice or, at the other end of the continuum, formal distributed classroom activities.

For example, a graph can represent a process composed of several imbricate procedures, their inputs and products. During delivery, it situates operations and resources in their context and facilitates the observation of the process. Generally speaking, the function model is used as a communication and control interface where a cause-effect loop appears: an actor can change his behaviour 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 reality takes place.

If the function model is analysed after the fact (a description of what took place) the gain 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 favourable direction. If the model is built during the action, in an emergent way, it embeds the necessary dynamism for life-long learning and professional development communities to build and develop, particularly in project-based learning situations.

Finally, because of their inclusion in learning object repositories, function models are themselves aggregated resources that can be referenced using metadata specification, then annotated, reused and interoperated with other functions to form complete distributed learning environments for the actors involved.

## 5. CONCLUSION

This article has presented delivery models enabling the design of actor-centred environments, aggregating resources to support the actor's operations within each function modelled when the system was designed.

We are currently upgrading the Explor@ system so that the graphic delivery models act as user interfaces at delivery time to facilitate the actors' interactions and co-ordination, a more general and dynamic alternative to the actual hierarchical menus or structures. This new version will display a graphic interface for each function of the learning system. It will inform the actors about the context of operations they perform in different functions, give access to the latest version of resources produced by other actors, and provide access to update resources for others. In addition, communication, metadata referencing, group annotation and assistance facilities will be accessible from the graphic object representing any operation.

We believe this solution will resolve many of the co-ordination difficulties encountered in all distributed learning systems. Especially in contexts where the actors and resources change regularly, the learners will benefit from constantly knowing where a specific resource or information can be found and which actors they can communicate with on that respect.

The actors' environments aggregate resources, which can be referenced using metadata standards. The models include assembly rules to build larger more meaningful resources by representing functions of the learning system. They aggregate actors, resources, functions, and environments dynamically. The results are new resources described not only by their component parts, their anatomy, but also by their dynamic aspects, their physiology.

Our future work will analyse impacts of this framework on the metadata referencing standards such as IMS. Wiley (2002) mentions that the main challenge for the interoperability of learning objects is in the instructional design rather than platform interoperability that has motivated their initial development. We also believe that instructional engineering is a key to offering solutions to aggregation and interoperability of learning objects.

I hope that the ideas presented in this article will contribute to the solution of the aggregation of learning objects, and more generally, to efficient and significant solutions to build more meaningful and useful distributed learning systems.

## REFERENCES

- Booch, G., Jacobson, J. and Rumbaugh, I. (1999) *The Unified Modelling Language User Guide*. Boston, MA: Addison-Wesley

- Harmbrech (2001) Corporate e-learning: Exploring new frontiers. Available from: WR Harmbrech + Co. [elarning@wrhambrecht.com](mailto:elarning@wrhambrecht.com)
- Koper, R. (2001) Modelling units of study from a pedagogical perspective – The pedagogical metamodel behind EML. (Available at <http://www.eml.ou.nl/introduction/articles.htm>)
- Lemoigne, J.L. (1995). *Les épistémologies constructivistes*. PUF : Que sais-je?
- McGraw, K.L. and Harbisson-Briggs, K. (1989) *Knowledge Acquisition*. Upper Saddle River, New Jersey: Prentice-Hall
- Merrill, M.D. (1994) *Principles of Instructional Design*. Englewood Cliffs, New Jersey: Educational Technology Publications
- Paquette, G. (1996) La modélisation par objets typés: une méthode de représentation pour les systèmes d'apprentissage et d'aide à la tâche. *Sciences et techniques éducatives*, 9-42
- Paquette, G. (1999) Meta-knowledge Representation for Learning Scenarios Engineering. In S. Lajoie and M. Vivet (eds.) *Proceedings of AI-Ed'99 in AI and Education, open learning environments*. IOS Press
- Paquette, G. (2001) TeleLearning Systems Engineering – Towards a new ISD model. *Journal of Structural Learning*, 14, 1-35
- Paquette, G., Aubin, C. and Crevier, F. (1999) MISA, A Knowledge-based Method for the Engineering of Learning Systems. *Journal of Courseware Engineering*, 2
- Paquette, G. and Rosca I. (unpublished) Modelling the Delivery Physiology of Distributed Learning Systems. Paper submitted to *Technology, Instruction, Cognition and Learning*, April 2002.
- Paquette, G., Rosca, I., De la Teja, I., Léonard, M. and Lundgren-Cayrol, K. (2001) Web-based Support for the Instructional Engineering of E-learning Systems. *WebNet'01 Conference*, Orlando.
- Paquette, G. and Roy, L. (1990) *Systèmes à base de connaissances*. Télé-université et Beauchemin, Montréal.
- Reigeluth, C. (1983) *Instructional Theories in Action: Lessons Illustrating Selected Theories and Models*. Hillsdale, NJ: Lawrence Erlbaum
- Schreiber, G., Wielinga, B. and Breuker, J. (1993) *KADS – A Principled Approach to Knowledge-based System Development*. San Diego: Academic Press
- Simon, H.A. (1981) *The sciences of the artificial*. Cambridge Mass: The MIT Press
- Spector, J.M., Polson M.C. and Muraida D.J. (eds.) (1993) *Automating Instructional Design, Concepts and Issues*. Englewood Cliffs, New Jersey: Educational Technology Publications
- 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

## BIOGRAPHY

**Gilbert Paquette** is director of the centre for interuniversity research on tele-learning applications and is a researcher at Télé-université's LICEF research centre which was founded in 1992. He holds a Canada research chair in knowledge-based instructional engineering.