Designer integration in training cycles : IEEE LTSA model adaptation

Alain Corbière, Christophe Choquet

▶ **To cite this version:**

Alain Corbière, Christophe Choquet. Designer integration in training cycles : IEEE LTSA model adaptation. CALIE 2004, International Conference on Computer Aided Learning in Engineering Education, 2004, Grenoble, France. 12 p. hal-00190308

HAL Id: hal-00190308

https://telearn.hal.science/hal-00190308

Submitted on 23 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DESIGNER INTEGRATION IN TRAINING CYCLES : IEEE LTSA MODEL ADAPTATION

Alain Corbière & Chistophe Choquet
(alain.corbiere@univ-lemans.fr, christophe.choquet@univ-lemans.fr)
LIUM – IUT de Laval
52, rue des docteurs Calmette et Guérin, 53020 LAVAL Cedex 9 - FRANCE
http://www-ic2.univ-lemans.fr/lium/

KEYWORDS: Re-engineering, Course cycle, Distance learning, Software architecture

Abstract

This paper describes an architecture centered on a component model for the course cycle. This model guides a re-engineering process based on the observed use scenarii. It is applied in an institutional framework and uses learning devices provided by the open source community. The architecture integrates the latest works on learning technology standards.

INTRODUCTION

The studies presented in this article are within the MOCA research project; one of its objectives is to re-engineer distance learning by integrating recent normalization efforts in educative technologies.

The international studies of the pedagogical resources such as the Learning Object Metadata analysis, as well as the distance learning platform architectures such as IEEE LTSA (Learning and Training System Architecture) are manifest today but are based essentially on diffusion. Few studies relate to the problems of resource design and their re-engineering within the distance learning and interoperability contexts on which the current normalization projects are based.

However, in an institutional framework (courses leading to a national diploma), the design and improvement of the pedagogical resources is assisted by the evolution of a traditional course (i.e. face to face teaching):

1. the teacher defines, creates, and organizes the contents into a pedagogical scenario that he finds functional
2. the teacher uses this material in class and modifies both the scenario and his own behavior as a result of the reaction and achievements of the students
3. he is therefore able to detect errors and see the possible improvements to his teaching
4. in most cases, this leads him to revise the contents and their sequence in preparation for the next class.

The use of Information and Communication technologies in the training process introduces a de-synchronization of the teacher’s two main roles : author/designer of the course and tutor/teacher. This leads to specialized actors in the training process today.

The process of re-engineering that we will present incorporates both the class itself and the different phases before and after to bring about an operationalisation of the course’s progression. It is based on the LTSA architectural model and the studies on the Learning Object Metadata (LOM) and the educational modeling languages (EML).

First, we will briefly describe the unified teaching process model proposed by the IEEE committee and its integration with the LOM proposal. We will conclude this part with an analysis of the EML potential in such an architecture. Then we will establish, on the basis of these studies, a training process within the normalized educational technologies environment and which facilitates re-engineering.

We will conclude this paper with a debate on the choices available that allows a rapid implementation of this re-engineering model.

LTSA : AN ARCHITECTURAL MODEL OF A LEARNING PROCEDURE

LTSA objectives

The IEEE committee for educational technologies is made up of both industrialists and researchers on an international level; it proposes an architecture (IEEE LTSA draft 11, 2002) resulting from information technology analysis. Without going into detail on the computing material necessary for implementation, this architecture gives a reference model for the different levels of the learning process :

(i) the context for the human actors

(ii) a component model for the organization and

(iii) the software development cycle.

The IEEE only mandates the central layer of the architecture, the “data flow diagram”. To validate this layer, the descriptive documentation is based on a typology of perspectives implemented in the learning process.

The architecture brings together the different roles of the work groups on the educative technologies future standards. For example, take the group that defines a metadata for a learning object. Its purpose is to respond to an interface requirement between the process controlling the learning process and the resources. The American (IEEE Metadata v1, 2002) and European (ARIADNE Metadata v3.2, 2002) groups are working closely together on this project.

The architecture is described below. It will be a means of formalizing the training process undertaken in our approach.

LTSA architecture description

The five different levels of the architecture represent the different points of view of a learning process (from the most abstract to the least).

Level 1 : This level is the most abstract and defines the tasks of acquisition, transfer, exchange and discovery for the learner as a result of the interactions with his environment. These environment and learner entities are seen as two systems exchanging information.

Level 2 : This layer, roughly detailed in the document, defines the learner’s reaction to the environment.

Level 3 : A component system, normalized by IEEE, defines an organization of a learning process seen from the data and control flow points of view.

Level 4 : This level exploits the component system directly in order to formalize the technological design constraints. It allows the identification of the system’s activities during the learning process.

Level 5 : This level defines the abstract phases of the software development based on the component approach.

A description of the LTSA system components (level 3)

The human and artificial actors involved in the functioning of a learning process are described in level three. In this model, four processes are highlighted : the learning, evaluating, tutoring and diffusion processes. In addition, two data stores save learner profiles and pedagogical resources.

The data and control flows link the different components; each one will lead to a standard interface.

The types of data exchanged in the data or control flows are defined by the IEEE's document 1484.1.

The *interaction context* : this flow of data gives the necessary information for the interpretation of the observations.

The *observations* : this data flow represents the real-time unabridged information concerning the learner activities.

The *acquisition state* : the evaluating process can send or update a learner profile (for example as a response to a correct answer within a given time).

The *learner profile* : through this data flow, the tutoring process can consult and modify learner information during the apprenticeship.

The *evaluation* : this data flow informs the tutoring process of the present state of the learner profile so as to optimize the learning process.

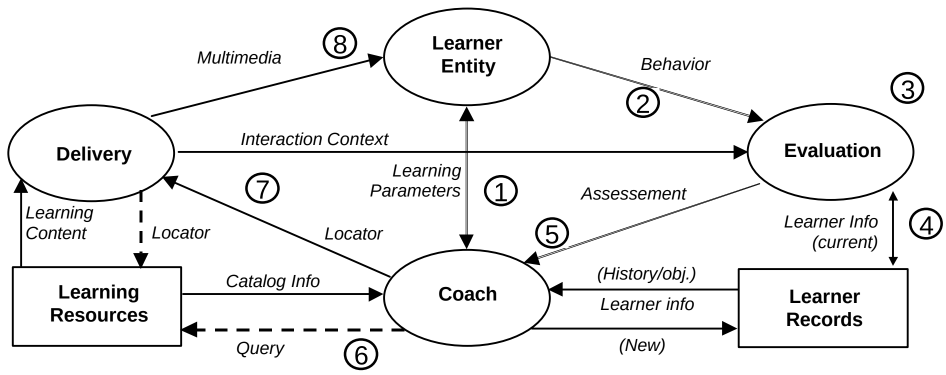
The *learner preferences* : the tutoring process negotiates the teaching parameters with the learning actor(s).

The *multimedia data* : this flow of data allows the learning process to use simultaneous pedagogical multimedia resources such as video, audio, text and graphs.

The *locality* : this data or control flow indicates where to find a given pedagogical resource.

The *pedagogical contents* : this data flow has the coded pedagogical material.

The *catalogued inquiries and information* : the tutoring process can carry out simple requests to find an appropriate learning object for a course. These requests may contain search criteria based on the learner's preferences, the evaluation results and the course information.



This figure respects the notations system defined by Yourdon in the e-book « Just Enough Structured Analysis ». (http://www.yourdon.com/books/msa2e/).

Figure 1: A component model for the learning system

The role and the behavior of the different components are described using a learner scenario, which is divided into eight steps :

1. The teaching style, the pedagogical choices, the acquisition methods are negotiated with the learner.
2. The learning process is observed and evaluated in a context of action and interaction with the system.
3. The evaluating process gives observations and indications about the learner style and/or information about the functioning or the state of the system.

4. This data is stored in a data bank dedicated to the learner.
5. The tutoring process analyses the learner's performance from his assessments, his preferences, his past history and his future perspectives.
6. This same process searches for suitable learning object using resource bank requests.
7. The tutoring process extracts the pedagogical content from the proposed resources. It transmits the resource references to the diffusion process, organizing them for example into a pedagogical sequence.
8. The diffusion process extracts the pedagogical contents from the learning object to adapt it to the surrounding interface used by the learner.

THE COURSE DESIGNERS EVOLUTION TO HELP THE PRODUCTION RE-ENGINEERING

LOM evolution

When associated with the LTSA architecture, a conceptual schema of the data (IEEE Metadata v1, 2002) can be used to describe and structure the content of a learning object. This structure is made up of nine groups (general, life cycle, meta-metadata, technical, educational, rights, relation, annotation and classification). It is a result of the adaptation of the widely acknowledged ‘‘Dublin Core Metadata Element Set’’ to the training technologies domain. This standard description tool allows the course actors to define the learning object metadata.

The designer, whether he is an ergonomist, a pedagogue or an educator, will enact strategies and knowledge. He needs to be able to integrate his expertise into the resource (Mizoguchi R., 1998). A large part of this knowledge is not exploitable with the help of metadata due to the priority given to the format description standardization. All the resource's pedagogical, educational, ergonomic or technical characteristics are constrained by the existing classifications. The descriptive cognitive capacities of a learning object are restricted so as to define it explicitly.

If we consider that the users of learning object make choices during the learning activity based on these descriptions, and require a number of guarantees for the user environment, the investment made to produce the resources, in terms of knowledge and savoir-faire, must be managed. In order to meet this goal in a training institution, the expertise shown by the different actors must be recognized and assessed.

This expertise capitalization seems a central question in the learning design engineering. We cite the works of (Bruillard E. et al., 1994) which give five methodical rules for design learning systems and which are based on a cross-disciplinary approach including the development of prototypes and both teacher and learner user assessment :

1. Starting from teaching problem and , if possible, an educational analysis;
2. Working within a team made up of computer scientists, educators and teachers from the very beginning of the project design;
3. Using an interaction situation model and building prototypes to establish the future system's specifications;
4. Assessing these prototypes as quickly as possible from both the teacher and learner points of view;
5. Centering the design around the learner-to-system interactions (or the student-student interactions) and using them to specify the learner's learning objectives and the learning circumstances;

In this way, even if the LOM corresponds suitably to the information system working constraints, it is insufficient to describe the numerous concepts pertaining to a learning object in its true working conditions, within a pedagogical scenario. The normalization committees are aware of this and are moving towards specific descriptive languages for e-learning such as EML (Rawlings A. et al., 2002).

The EML potential

The goal of such a language is not to highlight the information management constraints, but to describe as accurately as possible all the pedagogical aspects of a given resource. There are several conceptual models to describe a teaching unit using the role, activity and environment concepts. A number of these models are operational with XML technologies (De la passardière et al., 2001). A learning object is characterized by a LOM heading and within a number of scenarios described with EML. Designers use the language schema as a description means for the activities presented to the learner. The pedagogical objectives and potentials can de defined.

As an example, we could quote the Open University of the Netherlands (Koper R., 2001) which is one of the most advanced in its recent studies conducted by the European normalization group (Rawlings A. et al., 2002). This language describes a format independent scenario to present the different learner environments. The aim is to be able to adapt them to the chosen diffusion means. It corresponds well to the directives given by the normalization committee for educational technologies.

This tool allows the establishment of an *a priori* scenario. The conceptual model defining the descriptive language schema for a resource, or a group of resources, describes the activity proposed to the learner, the goals and the pedagogical potential.

It becomes therefore possible to integrate new conceptual elements which characterize a pedagogical scenario in a given situation. Thus, a designer may specify the exercise constraints for the different scenario elements with the aim of :

(i) detecting any disparities in the scenario during use and

(ii) allowing the adaptation of the scenario to the observed use (re-engineering)

As this process is iterative, a knowledge base making use of the expertise and experience of the designer community can be constituted. This favors the definition of *a priori* scenarii that are far better adapted to the expected usage.

This re-engineering and expertise capitalization process can only be formalized by taking the whole learning cursus into account and not just a single lesson. We will present an adapted LTSA model which includes this constraint.

AN ARCHITECTURAL MODEL OF A COURSE

Limits of the original LTSA model

The LTSA model does not regard the learning object designer as integrated in the learning process. He is outside the different levels of the model. The declared purpose is to be centered on the learning process and doesn't consider the other cycles of a distance learning course.

This limitation can no longer be justified when considering institutional frameworks, where courses are long and intended for multiple-year group learners.

By analyzing some projects resulting from the French working groups (RUCA[1]), we can see that the design process is integrated in the global system architecture from the first phases of structured analysis. We could mention the “Ulysse” project of Bordeaux University which is controlled by the CONCERTO 813 group (CONCERTO University group, 1997) under the

[1] Le Réseau Universitaire des Centres d’Autoformation – the French open university network

initiative of the RUCA. Additional processes, absent from the LTSA model, are integrated in the “Ulysse” project model. They highlight the necessity to integrate the actor-designer in the course process.

The LTSA component architecture does not intend to represent the entirety of the course process. Without going into detail on the organization and design aspects, the proposition of IEEE aims to standardize a unique learner-centered process model, intended for e-learning companies and integrating the standard characterization of a LOM meta-data (IEEE Metadata v1, 2002).

Moreover, pedagogical engineering defines the tasks of the designer following the psycho-educational and social sciences points of view. This shows that it is necessary to aware of difficult learning situations before the training session; this allows for improvement in the pedagogical scenario. The strategies and methods used to implement quality education integrate these tasks systematically.

For example, we can quote the contributions from the “HELICES” standard model for learner environment design. The designer uses this cyclic analytic framework to describe the actor, the task and the circumstances comprehensively and coherently (Linard M., 2001).

LTSA architecture evolution

To begin with, we propose a new stakeholder perspective implemented the LTSA system components. It is added to those presented in the annex C of the LTSA document. Subsequently, we will widen the spectrum covered by the initial model by integrating the « design » process. We will modify levels 1, 3 and 5 of the LTSA model to specify a new high-level architecture for information technology.

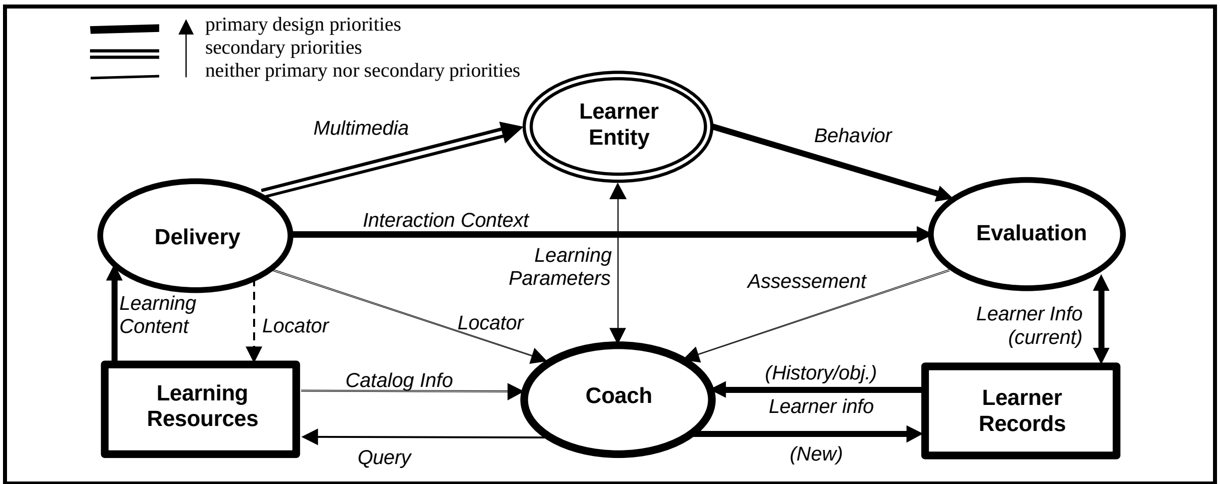
The significance of our approach is to use LTSA which is the result of a structured analysis carried out by the IEEE committee group since 1996. This new model provides a framework for the evaluation of the usage scenarii. The new components allow the analysis of the flows of information linked to the LTSA system’s component processes. During the training session, new operations must be carried out : tracking, contextualizing and storing digital tracks of the behavior system and interactions between human actors and the learning system.

New stakeholder mapping: Evaluation of the usage scenarios

The notations used to represent the figure 2 allow us to emphasize technology constraints with a particular behavior of the training system. This stakeholder, qualified as “overlapping”, adds new constraints to the other stakeholders referenced in the LTSA document.

The evaluation of the usage scenario is carried out by analysis functions of the data exchange between three processes (Delivery, Evaluation and Coach). Different tasks are carried out during the learning session: collecting digital tracks, converting to information and taking decisions. The designer describes these tasks in the learning scenario. The learning system, by implementing this perspective, must comply with this learning script. It must respect all the requirements expressed by the designer : from the semantics of the event collected by the leaning system component to the roles defined for each actor. The learner’s learning process will be analyzed later by the designer. All the decisions taken and the relevant data are stored for each learner profile.

This stakeholder “Evaluation of the usage scenario” supplements those presented in annex C and completes the suggested training device behavior typology in the document of IEEE.



Summary	To track the use of learning objects
LTSA Design Priorities	Primary : The organization and structure of the learning resources. The scope, functionality, user interface, and control inputs and outputs to the delivery process. The semantics, protocols, and formats of the learner information. The standards, procedures, methods, protocols, and formats of behavior observation. Secondary : The support tools accessible through the delivery process and sent via multimedia. The QoS of multimedia connections.
Non-LTSA Focus	Primary : The learner records store the learning scenarii uses. Secondary : The learner is free to choose his/her learning methods, style, and strategies.
Other Issues	Security methods to control access to the learner entity's history. Co-ordination and combination of various data mining systems.

Figure 2: New stakeholder mapping

Level 1 : Actor-environment interactions

This level represents the environment of the course cycle. We can distinguish the time spent on design from that for the training. In turn, this makes it possible to distinguish the design collaboration community from that of the learners.

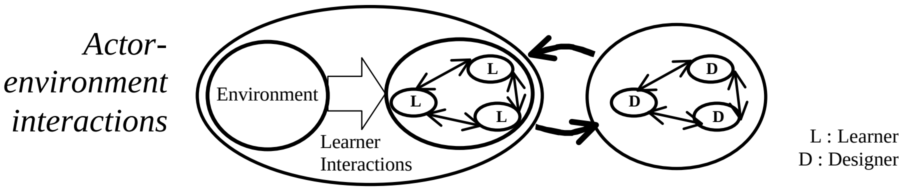


Figure 3: A system view of actor-environment interactions.

Level 3 : Re-engineering process component model

In level 3, we define a new component model to formalize the organization of distance course cycles in term of information flow. This model results from the structural analysis of the expertise carried out in each process.

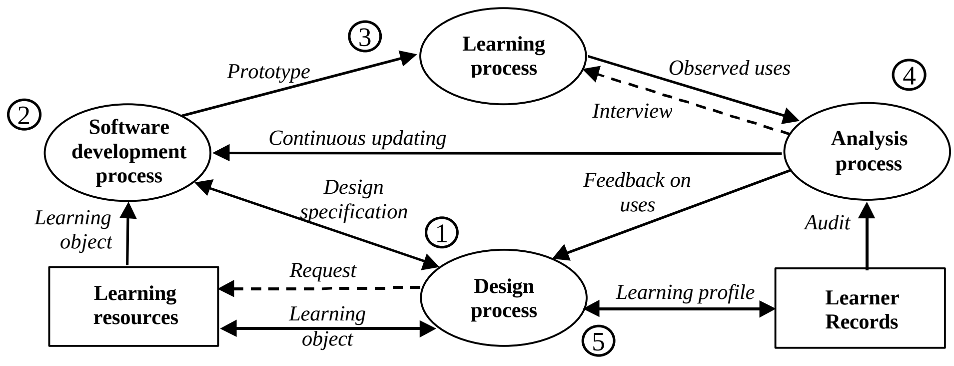


Figure 4: A component model for the course cycle

We use here the notation system introduced by Yourdon (Yourdon E. et al., 1979). The storage units are identical to those represented in the LTSA component model.

In the same way, following the interoperability standards, the data and control flows are detailed as follows :

Data-information flow:

Observed uses: The LTSA system components produce observations to define how the training activity takes place. This data information represents numerical tracks, interviews with learners and human coaches and the reactions of these different actors.

Feedback on uses: The analysis process highlights problems and patterns by data mining. Data is collected during the training session. This new information focuses attention on differences between the supposed use and that observed and is sent back to the designers for possible re-engineering of the scenario or the resources.

Continuous updating: Problems observed can be resolved simply by updating the contents and the structures of learning environments (Duquesnoy L. et al., 2002). The production process can directly manage these updates.

Learning profile: Via this data flow, the learning design process operates on learner information. Its reactions, following the use of a learning object, are integrated in its profile. This may also exemplify the object.

Design specification : The exchanges made define the activity sequence, the roles and the pedagogical environment. They include the observed uses. This data flow allows the specification of ergonomic interfaces and teaching sequences. It also describes the digital tracks. The learning and instructional designers refer to learner activities, learning environments and staff roles by using an EML.

Learning object: This refers to the learning environments which support the training process and generate the valid feedback. This object can be reused, modified or re-created from old objects. The designer community retrieves the objects validated after use.

Audit: This flow presents a history of observed uses during the previous learning processes. It’s the main input to the analysis process.

Prototype: The learning objects, in a re-engineering cycle, are considered as prototypes.

Control flow:

Interview: Using the observed uses, the analysis process appraises the scenario and learning object use. This flow completes the information obtained so as to guide the subsequent decisions.

Request: The design process can retrieve learning objects meeting its specific needs by a resource bank request.

The role and the behavior of processes are described in following sequence. We refer to the steps represented in figure 4. The design process (1) has to describe and define the academic training cursus in terms of scenario, training units and learning objects (Paquette G. et al., 1996). A set of modeling languages can be used to formalize this and, because of their guidance abilities, to help designers. Results, i.e. learner environment specifications and observation procedure interactions and uses, are provided in the software development process (2).

The software development process works from the specifications as defined by designers. They describe the entire set of resources needed by the learner environment. Learner session survey mechanism integration is one of the important tasks of the software development process. We advise developers to adopt a component oriented approach, so as to optimize the development phases. This respects the educational technologies standards recommendations for re-use.

The resources provided by the software development process are exploited by the learner process (3) whose architecture is precisely the component oriented architecture proposed by LTSA.

The analysis process (4) tries to correlate the *a priori* scenarii (i.e. defined by designers) with learner's uses and interactions with the environment. Inconsistencies can be detected by the learner's environment itself, or after a connection established by analysis between observed uses and the learner profile. Two courses of action may be taken. First, the software development process can modify the parameters of the survey mechanisms which are too reactive. If the training team thinks that the inconsistencies highlight some learners' critical problems; the second action can be carried out. For this, the data collected on the observed event context must be transmitted to the design process, in order to possibly modify (i.e. optimize) resources.

One of the design process (5) tasks consists in a pedagogical, ergonomic and didactic evaluation of the anomalies and feedback after a training session. Designers perhaps need to modify scenarii and resources, clarify some learning objects' metadata or be more specific in the way the system has to survey and guide the learner's activity.

This training cycle example shows clearly that the three processes (design, implementation and validation) have to take different decisions. These interventions can be carried out by human or software systems. They are especially founded on the interpretation and the understanding of the learner process actions. To guarantee the pertinence of these interventions, survey mechanisms have to be constantly evaluated. In this way, such mechanisms can become learning objects themselves, stored in the resource databank and described and qualified by a metadata.

Level 5: Re-engineering development process

Level 5 defines a guideline for learning institutes and information technology companies which produce and use learning components and management systems. Confronted with the profusion and diversity of information technologies, the learning manager using and integrating existing learning environments needs to have guarantees on the continuity of these technologies.

The IEEE Learning Technology Standards Committee proposes a generic development process to answer these concerns. It calls for the harmonization of information technology interoperability standards by minimizing the influence of the technologies used to implement the stakeholder perspectives. Each phase of this development process is guided by standard learning technologies (Farance F., 2000). After selecting a stakeholder perspective, generic task coordination is proposed in level 5.

This method compares to a unified process of development (Jacobson I. et al., 1999), and uses a unified modeling language (OMG UML, 2001). Both present iterative and incremental features. IEEE's is mainly centered on interoperability constraints whereas the unified process is centered on software architecture. On each abstraction level, an evaluation of the risks is carried out according to the choices of educational technologies and the requirements of the device actors. Both enter in a common re-use issue and adopt a component oriented process. At each step of the unified process, decisions are taken using and producing documents from the different points of view and from the different abstraction levels. But, the LTSA level 5 doesn't integrate the management of this information. However, level 4 is based on a stakeholder that defines the decisions taken with regard to the alternative technologies (Yourdon E. & al.,1979).

To perceive the course life cycle in its totality, different views on learning object must be exploited. Let us take the example of one cycle where the design process selects a learning environment and describes the use to be observed. At first, the software process starts with a reverse engineering task with an aim of highlighting the observable behavior and secondly performs the steps of the LTSA development process to integrate software sensors in the learning environment. In the learning sessions feedback, the analysis process interprets the tracks obtained compared to the technical documents provided by the actor software process.

Thereafter, the designer decides to integrate the description of a behavior, originally expressed in a technical modeling language, with an educational language.

New tasks of "reverse engineering" and "re-engineering", (Chikofsky E.J. et al., 1990), are added. We register the issue of software community maintenance in the context of educational technologies standards. The first task identifies the system components and their interactions and models representations of the software environment under different forms or on different abstraction levels. The second modifies existing components or creates new ones in the learning environment. Both tasks integrate process constraints by interoperability objectives. In the following figure, we adapt this proposed workflow by integrating new flows and new constraints for each step to perceive re-engineering development process cycles.

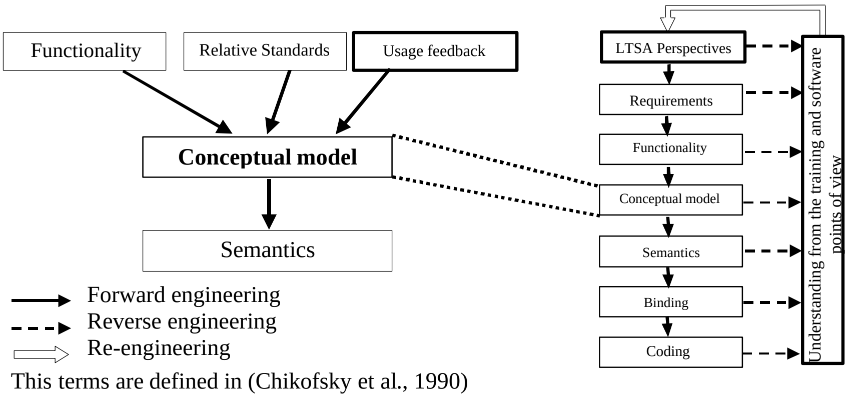


Figure 5: Evolution of workflow steps

Synthesis

In the integrated learning technology standards system, the varied levels of the students and effects on the learning contents defines a distance course as evolutive. We cannot consider the course cycle as a simple deposit of learning resource banks before a training session.

The design tasks are not only present to describe the learning environments and learning scenarii. We add analysis phases to consider the scenarii of observed uses in training sessions. In this article, we show distance course architecture integrating the LTSA model, which centers on the course re-engineering organization. This proposition represents an interaction model between the learning design process, the software development process, the learning process (represented by the LTSA system) and the analysis process. This is the first result of our project to compute the course re-engineering by knowledge management.

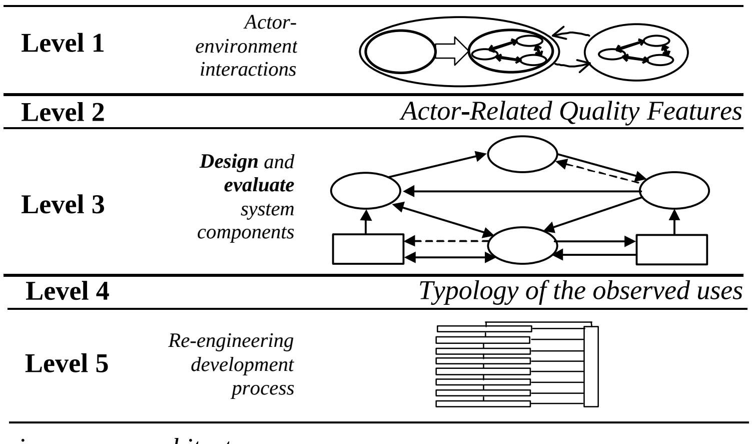


Figure 6: Distance learning course architecture

The intermediary levels 2 and 4 are not described in this article and are the subject of current work.

CONCLUSION

To propose a distance learning course process managed by cycles of re-engineering, a designer listens to and answers the learner requirements in a training session. It seems fundamental to us that the learning objects used must be seen as prototypes and not as finished products.

In the distance learning design, we can imagine the creation of practice community. The main objectives are the sharing of learning experience and the practice of distance learning sessions. The distance course architecture, described in this article, shows that the standardization endeavor contributes to the instigation of exchanges on feedback learning experiences and are considered thereafter as a learning experience bank.

From the start of our work the sequence of five steps, described in paragraph 4.2.3., is implemented within the framework of a course. We adhered to the open source software project for the learning platform. The one selected is composed of two entities: FreeStyle Learning (Brocke J.V. et al., 2000) and OpenUSS (Dewanto L., 2002). Thus, we can have an operational and evolutionary device. In this software project, the sharing of development resources facilitates the performance of the "reverse engineering" and "re-engineering" tasks.

Over three consecutive years, around fifty learners have used the learning device within a two-hour session. At each cycle, the observations on the course training session increase our expertise on the architecture representation and on the *posteriori* pedagogical scenario. It is interesting to question if the modeling languages in learning technology and in object oriented technology are able to describe and store this expertise. Thus, we search to describe (1) what makes sense to the learning designer actor in these modeling languages and (2) the most adapted inspection and interrogation methods with the learning design and re-engineering learning processes.

REFERENCES

ARIADNE Metadata v3.2 (2002). ARIADNE Educational Metadata Recommendation, version 3.2.EPFL (Lausanne, CH), K.U.Leuven (Leuven, B) and the ARIADNE Foundation.

BROCKLE J. V. & LAHME N. (2000). Freestyle Learning A new way to Computer Assisted Learning (CAL). in: WEM Paper, World Education Market, Vancouver 2000.

BRUILLARD E. & VIVET M. (1994). Concevoir des EIAO pour des situations scolaires : approche méthodologique, didactique et Intelligence Artificielle. La pensée Sauvage, Grenoble : 273-302.

CHIKOFSKY E. J. & CROSS II J. H. (1990). Reverse Engineering and Design Recovery : A taxonomy. IEEE Software, volume 7, number 1: 13-17.

CONCERTO University group (1997). Système informatisé d’autoformation et d’autoévaluation formative à l’aide de ressources pédagogiques multimédias. Sciences et techniques éducatives. Volume 4, number 4. Hermes editions.

DE LA PASSARDIERE B. & GIROIRE H. (2001). XML au service des applications pédagogiques, Actes des journées EIAO’2001. Sciences et Techniques Educatives. Hermes editions. Volume 8: 99-112.

DEWANTO L. (2002). ObjectWeb and OpenUSS: a win-win situation?. Conference objectweb 2002, INRIA Rocquencourt.

DUQUESNOY L., BERGER J.-L., PREVOT P. & SANDOZ-GUERMOND F. (2002). SIMPA: A Training Platform in Work Station Including Computing Tutors. 6th International Conference ITS 2002, Biarritz.

FARANCE F. (2000). Work Program and Process of Learning Technology Standards Committee of the IEEE meeting, London, England.

IEEE LTSA draft 11 (2002). Draft Standard for Learning Technology Systems Architecture (LTSA). Working Paper of Learning Technology Standards Committee of the IEEE.

IEEE Metadata v1 (2002). Draft Standard for Learning Object Metadata (LOM). Learning Technology Standards Committee of the IEEE.

IMS Learning Design v1.0 (2002). IMS Learning Design Information Model. Version 1.0 Final Specification.

JACOBSON I., BOOCH G. & RUMBAUGH J. (1999). The Unified Software Development Process. Addison-Wesley Pub.

KOPER R. (2001). Modeling units of study from a pedagogical perspective, the pedagogical meta-model behind EML. First Draft, version 2. Open University of the Netherlands.

LINARD M. (2001). Concevoir des environnements pour apprendre : l'activité humaine, cadre organisateur de l'interactivité technique. Sciences et techniques éducatives. Hermes editions. Volume 8, numéro 3-4: 211-238.

MIZOGUCHI, R. (1998) A Step Towards Ontological Engineering. 12^{th} National Conference on AI of JSAI, pp.24-31, June, 1998.

OMG UML (2001) Unified Modeling Language (UML) OMG specification, v1.5.

PAQUETTE G., RICCIARDI-RIGAULT C., PAQUIN C., LIEGEOIS S. & BLEICHER E. (1996). Developing the Virtual Campus Environment. ED-Media Internationale Conference, Boston, USA.

RAWLINGS A., ROSMALEN P., KOPER R., RODRIGUEZ-ARTACHO M. & LEFRERE P.(2002). Survey of Educational Modelling Languages (EMLs), version 1, CEN/ISSS WS/LT Learning Technologies Workshop.

YOURDON E. & CONTANTINE L. (1979). Structured Design. Englewood Cliffs, NJ: Prentice Hall.