



**HAL**  
open science

## Scénarisation pédagogique et modèles conceptuels d'un EIAH : Que peuvent apporter les langages visuels?

Thierry Nodenot

► **To cite this version:**

Thierry Nodenot. Scénarisation pédagogique et modèles conceptuels d'un EIAH : Que peuvent apporter les langages visuels?. *Revue internationale des technologies en pédagogie universitaire*, 2007, 4 (2), pp.85-102. hal-00343617

**HAL Id: hal-00343617**

**<https://hal.science/hal-00343617>**

Submitted on 2 Dec 2008

**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.

---

# Scénarisation pédagogique et modèles conceptuels d'un EIAH : que peuvent apporter les langages visuels?

---

**Thierry Nodenot**

Laboratoire LIUPPA, IUT de Bayonne, FRANCE

---

Scénariser, une activité de formation

---

## Résumé

Cet article examine les apports des langages visuels pour la scénarisation pédagogique d'un EIAH. Basée sur l'évaluation de différents langages visuels, notre étude examine les modèles conceptuels que ces langages permettent de produire.

Plusieurs apports sont identifiés et discutés : les langages visuels sont des outils exploratoires qui aident les concepteurs à définir les contours d'un EIAH; ces langages permettent aussi aux différents acteurs d'échanger des points de vue au sein d'une équipe de développement pour aboutir à une solution viable.

La discussion souligne qu'au-delà de leurs aspects graphiques, l'expressivité des langages visuels devrait s'appuyer sur trois piliers essentiels : des primitives de conception tangibles pour les acteurs impliqués dans l'équipe de développement, une capacité à exprimer différentes perspectives d'un EIAH et différents niveaux d'abstraction pour ces perspectives.

## Abstract

In this paper, we study the added-value of visual languages when used to design the learning scenario of *technology-enhanced learning* (TEL) systems. We particularly focus on the conceptual models that designers can produce from such visual instructional design languages.

Several benefits are identified and discussed. First, such visual languages are well fitted to help designers explore the requirements of TEL systems; they also enable designers exchanging different viewpoints in order to reach a viable technical solution.

The discussion section pinpoints that beyond their graphic features, visual instructional design languages should conveniently mix adequate design primitives with stratification and multi-perspectives capabilities.

## Introduction

Reprenant un texte qu'il avait écrit en 2000, Baker (2006) a montré qu'il était aujourd'hui nécessaire de réétudier les finalités des trois types de modèles utilisés dans le champ des environnements informatiques pour l'apprentissage humain (EIAH) : le modèle comme composant logiciel de l'EIAH, le modèle pour l'évaluation des apprentissages humains et le modèle comme fondement pour la conception de ces systèmes. Tout comme pour les deux autres types de modèles, la perception de M. Baker semble avoir évolué sur la problématique des modèles de conception des EIAH. Baker (2006) souligne en effet que les modèles conceptuels d'un dispositif pédagogique (ou EIAH) doivent rester *ouverts, délibérément et stratégiquement imprécis, non linéaires et non déterministes puisqu'ils doivent se fonder sur un dialogue entre spécialistes différents*.

Afin de préciser l'organisation et le déroulement des situations d'apprentissage proposées par un dispositif pédagogique, les concepteurs établissent souvent une forme spécifique de modèle conceptuel : le scénario pédagogique. L'activité de scénarisation des apprentissages est donc étudiée dans de nombreux travaux de recherche du point de vue tant :

- de la précision, de la linéarité, du caractère plus ou moins déterministe des scénarios produits (Dillenbourg et Tchounikine, 2007), que
- des langages permettant d'exprimer ces scénarios et éventuellement de les exécuter sur des environnements cibles tels que des plateformes de formation ouverte et à distance. Parmi eux, citons le langage IMS-LD (IMS Global Learning Consortium, 2003a, 2003b) et le langage LDL (Ferraris et Martel, 2007).

En tant que domaine de recherche, les langages visuels font l'objet de nombreux travaux tant du point de vue de la théorie (Marriott et Meyer, 1998) que sur le plan de leurs usages en modélisation de systèmes ou programmation d'applications (Burnett, 2007). L'intérêt d'un langage visuel est d'offrir des notations pour faciliter la production/la compréhension

d'un modèle conceptuel, chaque représentation visuelle de ce modèle conceptuel étant basée sur l'une ou l'autre des notations offertes par le langage.

Cet article étudie plus spécifiquement les apports des langages visuels à la problématique des modèles de conception des EIAH. Après un rapide tour d'horizon des modèles que ces langages permettent de produire (*cf.* partie 2), nous détaillons (*cf.* partie 3) les apports potentiels et le champ d'application de ces langages visuels en basant notre étude sur quelques invariants qui ressortent des propositions récentes. La partie 4 nous permet de dresser un bilan, puis de proposer quelques pistes de recherche visant à généraliser l'utilisation de ces langages au-delà des niches de concepteurs qui les utilisent actuellement.

## Les représentations visuelles utilisées en conception des eiah

Les langages visuels au service de la scénarisation pédagogique fournissent des notations pour représenter les résultats intermédiaires et finaux d'un processus de conception qui peut s'apparenter à un processus d'ingénierie des connaissances (Paquette, Léonard, Lundgren-Cayrol, Mihaila et Gareau, 2006). Ces langages visuels peuvent donc servir des objectifs assez différents.

- Les uns se focalisent sur la représentation d'un ou plusieurs aspects du processus d'apprentissage/d'enseignement avant même sa mise en œuvre (Motschnig-Pitrik et Dertnl, 2005);
- D'autres cherchent davantage à expliciter les contenus de formation mis à disposition des apprenants.

De nombreux travaux de recherche ont porté sur le second type de langage, l'objectif étant de représenter formellement des unités d'apprentissage pouvant être transformées en code exécutable par une machine. Au contraire, assez peu de langages visuels ont été mis au point pour supporter la créativité des acteurs qui tentent de définir et de modéliser de manière fine des processus d'apprentissage et d'enseignement (Botturi, Cantoni, Lepori et Tardini, 2006).

**Tableau 1.** Les principaux types de représentations visuelles offerts par les langages étudiés

	E <sup>2</sup> ML	CPM	coUML	MOT+	PoEML	ASK-LDT	COLLAGE
Rôles et responsabilités		+ (cf. figures 2 et 3)	+		+	+	+
Modalités d'apprentissage	+		+ (cf. figure 4)			+	
Domaine, connaissances	+	+	+	+ (cf. figures 5 et 6)			
Objectifs et buts d'apprentissage	+ (cf. figures 7 et 8)	+		+			
Structure du cours	+			+	+	+ (cf. figure 9)	
Collaborations entre acteurs	+	+	+			+	+ (cf. figure 10)
Liens avec l'infrastructure		+	+	+	+ (cf. figure 11)	+	

Tout au long de cet article, nous traitons uniquement des langages visant à supporter la créativité des concepteurs en phase de modélisation des processus d'apprentissage et d'enseignement. Les langages pris en compte par notre étude sont les suivants :

- Pour les langages sans lien direct avec IMS-LD : E<sup>2</sup>ML (Botturi, 2003), CPM (Laforcade, Nodenot et Sallaberry, 2005), coUML (Derntl et Motschnig-Pitrik, 2007), qui reprend les travaux menés sur PCeL (Derntl, 2005), et PoEML (Caeiro-Rodriguez, Llamas-Nistal et Anido-Rifon, 2006);
- Pour les langages et éditeurs proches d'IMS-LD, nous évoquerons les caractéristiques de MOT+ (Paquette *et al.*, 2006), COLLAGE (Hernández-Leo *et al.*, 2006) et ASK-LDT (Sampson, Karampiperis et Zervas, 2005) en nous basant sur la synthèse faite dans Griffiths et Blat (2005).

Nous proposons tout d'abord une synthèse des représentations visuelles que ces langages permettent de produire. En effet, l'étude menée nous a conduit à identifier des invariants dans les représentations proposées, comme indiqué au tableau 1. Pour chaque représentation visuelle identifiée, le tableau ci-dessous liste les différents langages offrant ce type de représentation (cf. les symboles + apparaissant dans le

tableau) et propose un renvoi vers un exemple de représentation visuelle jugé représentatif des possibilités de ces langages (l'ensemble des exemples de représentations visuelles a été rassemblé en annexe de cet article). Il est bien évident que pour un type donné de représentation visuelle, des différences existent entre les langages étudiés et nous renvoyons le lecteur vers les références bibliographiques pour un examen plus détaillé de chacun de ces langages visuels.

Comme indiqué au tableau 1, nous avons identifié sept types de représentations visuelles pour les langages étudiés :

- La représentation des rôles et responsabilités que prennent les acteurs dans les situations d'apprentissage décrites. Au-delà de la simple différenciation apprenant/tuteur, la plupart des langages permettent de décrire, pour chaque module d'enseignement, une hiérarchie de rôles contextualisés en fonction de l'objet même du module (cf. figures 2 et 3);
- La représentation des modalités d'apprentissage prescrites pour chaque module d'enseignement identifié. Ces modalités (cf. figure 4) couvrent le déroulement dans le temps des activités du module, la synchronisation des activités conduites par différents acteurs et la différenciation des activités d'apprentissage médiatisées par l'outil informatique

de celles qui ne le sont pas (cf. les apprentissages de type *blended learning*);

- La représentation des connaissances sous-jacentes aux activités d'apprentissage prescrites. Cette description, souvent délaissée dans les travaux de scénarisation pédagogique, était prépondérante dans tous les travaux d'ingénierie des tuteurs intelligents (cf. la dichotomie entre le modèle du domaine, le modèle du tutorat, le modèle de l'apprenant, le modèle d'interface). Le tableau 1 montre que plusieurs langages visuels proposent des notations permettant de relier les activités prescrites aux concepts du domaine, aux savoirs et savoir-faire liés à ce domaine (cf. figures 5 et 6);
- La représentation précise des buts d'apprentissage (Gronlund, 1995) qui, pour certains langages visuels, doit rester conforme à des modèles théoriques tels que décrits dans Gagné, Briggs et Wager (2005) ou Merrill (1994). Ces buts associés (sous forme de prérequis/postrequis) aux activités d'apprentissage (cf. figures 7 et 8) sont souvent décrits selon différents niveaux d'abstraction;
- La représentation de la structure d'un module d'apprentissage sous forme d'activités reliées les unes aux autres par des relations de dépendance fonctionnelle ou de composition (cf. figure 9);
- La représentation des collaborations synchrones/asynchrones entre acteurs participant à une situation d'apprentissage. Cette représentation permet d'assigner aux acteurs leurs responsabilités respectives, les différentes collaborations à établir pouvant être spécifiées au cours du processus de conception d'une situation d'apprentissage particulière (cf. figure 4) ou préexister sous la forme d'un patron de conception (cf. figure 10);
- La représentation de la façon dont les activités d'apprentissage ainsi prescrites vont finalement être regroupées en paquetages (cf. figure 11) puis mises en œuvre sur une infrastructure logicielle donnée grâce aux composants et services fournis par cette infrastructure (modèle de diffusion).

Du point de vue de la théorie des langages (Cook, 2002; Ferruci, Tortora et Vitello, 2002), chacune de ces représentations ne peut s'analyser du seul point de vue visuel, mais en tant que modèle produit grâce à l'une des notations offertes par

le langage visuel utilisé. En effet, Steve Cook identifie trois aspects principaux d'un langage de modélisation : la syntaxe concrète, la syntaxe abstraite et la sémantique : la syntaxe concrète peut être définie sous la forme d'une grammaire, d'un dialecte XML ou d'un ensemble de notations graphiques telles que celles qui sont utilisées aux figures 2 à 11; ces notations graphiques représentent les concepts du langage tels que définis par la syntaxe abstraite de ce langage. Enfin, la sémantique définit la signification des modèles.

L'ensemble des représentations visuelles complémentaires produites pour un EIAH particulier constitue donc le modèle conceptuel de cet EIAH. Aussi, l'étude de la syntaxe abstraite et celle de la sémantique sont des éléments importants pour apprécier les apports possibles d'un langage visuel lors de la conception d'un EIAH. Ces éléments permettent d'évaluer notamment le positionnement de ce langage par rapport à un type de situation d'apprentissage à décrire (par exemple, les EIAH favorisant une pédagogie du projet, les EIAH privilégiant des simulations interactives pour des apprentissages de type *inquiry learning*, etc.) ou par rapport au cycle de vie d'un EIAH (des langages plus ou moins proches des fonctionnalités offertes par les infrastructures logicielles utilisées pour implémenter les EIAH). Dans la prochaine partie, nous allons poursuivre notre étude des sept langages sélectionnés en examinant leurs apports respectifs au cours du processus de conception d'un EIAH.

## Que peuvent apporter les langages visuels?

Reprenant les travaux de plusieurs auteurs, Stubbs et Gibbons (2007) font apparaître trois grandes phases de conception selon le destinataire des modèles visuels produits. Le processus de conception démarre la plupart du temps par 1) une période exploratoire (cf. *self-communication*, *graphic ideation*) permettant au concepteur de se représenter à grands traits une idée (un processus d'apprentissage, une interaction) pour mieux la comprendre et en délimiter les contours. Vient ensuite 2) une période d'approfondissement (cf. *developmental drawing*, *interprofessional communication*) qui, par une série d'approximations, permet d'incorporer des informa-

tions plus concrètes aux schémas et diagrammes précédents jusqu'à aboutir à une forme stabilisée et acceptée par les différents acteurs. C'est donc au cours de cette période que les modèles de conception peuvent être partagés avec d'autres concepteurs. Au fur et à mesure que ces modèles visuels sont affinés, ils deviennent plus formels et cessent d'avoir pour ambition de clarifier des idées pour devenir des propositions à destination d'interlocuteurs extérieurs : c'est 3) une période de préparation à la diffusion (cf. *client/public communication*).

Dans les prochains paragraphes, nous examinerons les possibilités offertes par les langages visuels que nous avons choisis pour chacune des deux premières phases citées.

### Des outils exploratoires

De belles idées naissent parfois d'un schéma griffonné sur un bout de papier et plusieurs auteurs considèrent que ce caractère informel et inaccompli présente un avantage au début du processus de conception pédagogique. Qu'il s'agisse de délimiter les contours de la situation d'apprentissage ou les spécifications d'un EIAH au service de cette situation d'apprentissage, faire un schéma informel, c'est pour de nombreux auteurs utiliser une représentation adéquate, car adaptée au caractère mal défini des problèmes de conception : « *It is widely accepted that design problems can only be regarded as a version of ill-defined problems* » (Cross, 2001). À l'opposé, il n'est pas raisonnable, comme le fait le guide des bonnes pratiques d'IMS-LD, de considérer que la conception commence lorsque les enseignants ont exprimé un scénario pédagogique de type *formal narrative* (IMS Global Learning Consortium, 2003a). *Les langages que nous avons étudiés sont un moyen terme entre ces deux positions puisqu'ils proposent un formalisme semi-formel visant à catalyser les idées des concepteurs*. Citons :

- les auteurs du langage E<sup>2</sup>ML qui, pour démarrer le processus de scénarisation, proposent l'utilisation conjointe de diagrammes temporels pour séquencer les activités d'apprentissage décrites (ces diagrammes sont proches de ceux produits à la figure 4 avec coUML) et de diagrammes de structure (proches des diagrammes de dépendances UML) pour agréger ces activités en paquetages;
- les auteurs du langage CPM qui ont choisi de faire aborder la conception par la production de diagrammes de *use cases* et de diagrammes de classes au sens UML (cf. figures 2 et 3). Ceci permet de définir rapidement les responsabilités des acteurs impliqués (éventuellement de manière collective) dans les différentes parties du scénario pédagogique;
- les auteurs du langage MOT+ qui proposent comme point de départ de la scénarisation (cf. la phase 2 de la méthode MISA) de faire exprimer aux concepteurs un modèle graphique des connaissances en relation avec la situation d'apprentissage étudiée (cf. figures 5 et 6). Les schémas au sens de Holoyak (1991) sont constitués de trois types de boîtes représentant les concepts du domaine, les principes et les procédures. Ces boîtes sont reliées dans un diagramme par des liens de type spécialisation, composition, instanciation, précedence et intrant/produit.

Une autre caractéristique importante des langages visuels étudiés est qu'ils proposent différentes perspectives de modélisation aux concepteurs pour un même objet modélisé. Pour expliquer la portée de cette approche, Gibbons et Stubbs (2007) citent les travaux de McKim postulant que le formalisme imposé par un langage (ou une perspective) est un support important pour guider la réflexion d'un concepteur, mais surtout que les concepteurs sont amenés à recentrer leur activité de réflexion par le fait même de passer d'une perspective à une autre pour étudier le problème de conception posé. Tous les langages étudiés proposent différents types de perspectives pour réduire la charge cognitive du concepteur. Citons tout d'abord le langage PoEML qui propose douze perspectives dont certaines doivent nécessairement être renseignées (objectifs pédagogiques, participants, regroupement structurel des activités, environnement dans lequel les activités sont déployées) et d'autres sont optionnelles : séquençage temporel des activités, fonctionnalités d'outils offertes aux apprenants, etc. Une focalisation particulière est portée sur la prise en compte de la modélisation des activités coopératives (droits des acteurs sur les fonctionnalités d'outils, événements liés à la conscience de groupe, règles régissant les interactions au sein d'un groupe d'apprenants).



Pour sa part, le langage coUML propose trois types de perspectives. Des perspectives qu'il est nécessaire de renseigner (*course activity model*, *course structure model*); des perspectives complémentaires permettant de préciser les rôles des acteurs, les buts pédagogiques et les documents exploités; et enfin une perspective dite auxiliaire (*course package model*) qui permet d'agrèger les autres perspectives en fournissant un point d'entrée pour la conception.

Le langage MOT+ propose quant à lui quatre perspectives : le modèle de connaissances déjà cité, le modèle de transmission (modèle pédagogique), le modèle des ressources nécessaires à un cours (modèle médiatique) et le modèle de diffusion.

*Les langages étudiés essaient enfin de mettre les concepteurs en situation authentique en proposant des primitives proches de leurs préoccupations pédagogiques; l'intérêt étant que les concepteurs peuvent s'appropriier plus facilement ces primitives pour construire les artefacts mis en lumière durant l'activité de scénarisation pédagogique.*

Les primitives d'un langage constituent la syntaxe abstraite de ce langage. Elles sont parfois choisies en référence à la métaphore privilégiée par le langage; c'est le cas de certaines primitives (*act*, *play*) des langages visuels s'appuyant sur IMS-LD qui découlent de la métaphore théâtrale proposée par IMS-LD. D'autres métaphores sont envisageables bien qu'elles ne soient pas forcément exploitées par les langages visuels étudiés dans cet article. Citons notamment la métaphore cinématographique (*cf.* les langages E<sup>2</sup>ML ou PoEML) qui permet de structurer un scénario sur la base d'un déroulement temporel (*timeline*), ou encore la métaphore des pléiades (Villiot-Leclercq et David, 2007) récemment proposée pour agréger les activités d'apprentissage selon des relations de précédence, hiérarchique, etc.

Mais notre étude nous a conduit à constater que, finalement, assez peu de concepts des langages étudiés dépendaient réellement de la métaphore mise en avant par ces langages. Parmi les primitives structurantes des langages étudiés, citons celles proposées par MOT+ à base de concepts (types

d'objets, de documents, d'outils, d'événements, de personnes), de procédures (opérations, tâches, activités ou scénarios) et de principes (propriétés des concepts, contraintes, relations de cause à effet, etc.).

Les primitives d'E<sup>2</sup>ML (*cf.* figures 7 et 8) reprennent la terminologie du modèle scientifique proposé par Anderson et Krathwohl (2001), à savoir une typologie des objectifs d'apprentissage (fait, concept, procédure, principe, attitude, etc.), un degré de perception de ces objectifs en quatre niveaux (expérience, compréhension, prise de recul, disponibilité) et une métrique identifiant la façon dont l'apprenant peut mobiliser ces objectifs (simple mémorisation, utilisation dans des situations spécifiques, exploitation pour générer de nouvelles connaissances).

Les primitives proposées par CPM reprennent les concepts de l'apprentissage à base de situations problèmes (objectifs, registre de conceptualisation, expression des représentations des apprenants, obstacles anticipés, activités de remédiation et de renforcement, etc.).

L'outil ASK-LDT propose au concepteur des primitives qui sont autant de blocs de construction mettant en relation les actions respectives des apprenants et des enseignants. Ces blocs (*cf. exploration/documentation, experimentation/reactivity, debate/animation, self-reflection/co-reflection, etc.*), qu'il s'agit de contextualiser dans le cadre de l'activité de scénarisation, sont issus du modèle 8LEM (Verpoorten, Poumay et Leclercq, 2006).

Nous citerons enfin les primitives offertes par COLLAGE qui propose des blocs de construction sous forme de patrons de coopération (par exemple, le patron *pyramid*, le patron *jigsaw* ou le patron *think-pair-share* décrit à la figure 10) qu'il s'agit là encore de contextualiser et d'assembler au service d'une situation d'apprentissage concrète.

### **Des outils pour approfondir des idées avec d'autres concepteurs**

Comme le notent Stubbs et Gibbons (2007), il arrive un moment où les modèles exploratoires qui sont produits ont tendance à se stabiliser, l'idée étant devenue suffisamment mûre pour être partagée avec d'autres concepteurs, notamment ceux qui ont une expérience dans la réalisation d'environnements informatiques capables d'exploiter les scénarios d'apprentissage décrits. L'équipe de conception entre alors dans une phase où il s'agit d'approfondir les modèles (notamment en définissant différentes perspectives) issus de la phase exploratoire pour, d'une part, détailler les aspects opérationnels envisagés et, d'autre part, confronter/adapter ces aspects opérationnels avec les fonctionnalités offertes par les infrastructures techniques choisies pour leur mise en œuvre.

*Pour permettre à une équipe de concepteurs de mener ce type de travail, certains langages visuels offrent différents niveaux d'abstraction.* Les gains d'une structuration en couches d'un modèle sont reconnus, chaque niveau d'abstraction pouvant apporter de l'information plus ou moins détaillée, tout en restant cohérente avec les niveaux supérieurs. Gibbons (2003) a identifié sept niveaux d'abstraction pour les applications éducatives. Il distingue les couches *content*, *strategy*, *control*, *message*, *representation*, *media logic* et *data management* qui sont reliées les unes aux autres. Ainsi, concevoir au niveau de la couche *control*, c'est déterminer les moyens d'accès dont dispose l'apprenant pour interagir avec les contenus de l'apprentissage; alors que concevoir au niveau de la couche *message*, c'est déterminer les types de messages à vocation pédagogique échangés avec l'apprenant, de quoi ils sont composés et comment ils sont générés.

La stratification aide donc une équipe de conception à déterminer les relations entre les niveaux d'analyse, permettant ainsi à chaque concepteur de situer son regard à un niveau donné (plus pédagogique ou plus fonctionnel/technique) tout en assurant une cohérence globale au scénario. Parmi les langages visuels que nous avons étudiés, seuls certains offrent différents niveaux d'abstraction aux concepteurs. Citons le langage PoEML qui permet d'étudier les douze perspectives citées dans la sous-section précédente selon quatre niveaux d'abstraction (présentés ici du plus concret au plus

abstrait). Le niveau le plus élémentaire porte sur les données manipulées (cf. les couches *data management*, *media logic* et *representation* du modèle proposé par A. Gibbons), c'est-à-dire des éléments de données simples (booléens, entiers, etc.) ou structurés (fichiers, pages Web, etc.) permettant d'instancier les objectifs d'apprentissage (prérequis et postrequis), les participants (données relatives à leur profil) et les productions faites par les participants grâce aux outils mis à leur disposition. Le deuxième niveau porte sur la gestion de l'exécution du modèle en fonction de certaines conditions ayant trait aux valeurs des données (cf. la couche *message* du modèle de A. Gibbons). Le troisième niveau porte sur le contrôle de l'exécution du scénario non pas en fonction de conditions sur les données et messages échangés, mais en fonction des choix (décisions) de participants ayant un niveau de responsabilité donné dans l'exécution du scénario (cf. la couche *control* du modèle proposé par A. Gibbons). Le dernier niveau porte sur l'apparition d'événements significatifs qui, lorsqu'ils se produisent, requièrent des changements stratégiques dans la mise en œuvre d'un scénario (ex. : lorsqu'on détecte que l'apprenant éprouve des problèmes ou qu'il ne prend pas sa place dans les activités collectives qui lui sont proposées).

Les langages CPM et coUML permettent également d'analyser un scénario à différents niveaux d'abstraction. Ces deux langages visuels qui sont construits comme des spécialisations du langage UML reprennent les mécanismes offerts par UML pour couvrir le processus de conception Fowler (2003) depuis la création de schémas exploratoires (cf. *UML as sketch*) jusqu'à la création de diagrammes conceptuels plus détaillés et plus formels portant sur des aspects particuliers des modèles, ceci afin de pouvoir étudier le développement d'une solution opérationnelle (cf. *UML as blueprint*). Ces deux langages proposent un ensemble de stéréotypes spécifiques (exemple : le stéréotype *ressource* du langage CPM, le stéréotype *blended activity* de coUML) qui peuvent être insérés dans des diagrammes de *use cases*, des diagrammes de classes, mais aussi des diagrammes d'activité au sens UML. Ces derniers sont très utiles pour décrire le séquençement des activités, détailler les ressources échangées entre acteurs au cours de ces activités, identifier leurs conditions d'exécution



**Tableau 2.** Les critères de classification des langages de modélisation pédagogique et le positionnement de certains langages selon Botturi, Derntl *et al.* (2006)

	Niveau de stratification	Niveau de formalisation	Niveau d'élaboration	Nombre de perspectives	Notation
E <sup>2</sup> ML	plat	semi-formel	conceptuel	plusieurs	visuel
PCeL	plusieurs couches	semi-formel	conceptuel	une	visuel
AUTC	plat	informel	spécification	plusieurs	visuel
IMS-LD	plusieurs couches	formel	spécification	une	textuel
PoEML	plusieurs couches	formel	implémentation	plusieurs	visuel
UML	plusieurs couches	formel	conceptuel / spécification	plusieurs	visuel

en fonction des valeurs prises par les ressources ou des résultats des activités antérieures, etc. Les diagrammes d'états UML sont également utilisés par ces langages pour exprimer notamment le cycle de vie des ressources et des activités ainsi que les événements significatifs des concepts décrits.

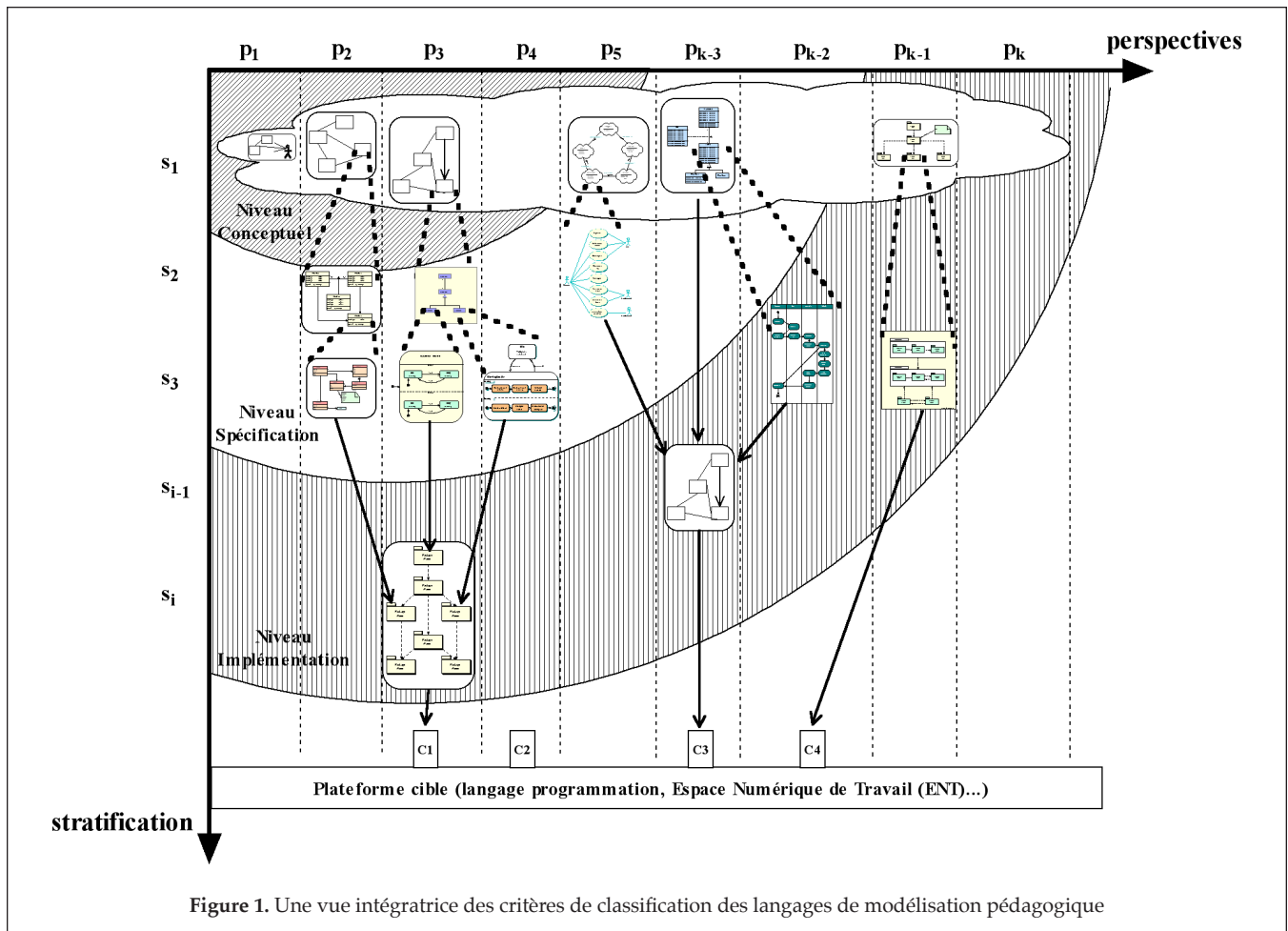
Pour le langage CPM (Nodenot, Laforcade et Le Pallec, 2007), le processus de conception démarre par la production de diagrammes de *use cases*. Chaque diagramme est ensuite raffiné par d'autres diagrammes de *use cases* puis par un, voire plusieurs diagrammes d'activité afin de décrire ce qui se déroule à l'intérieur d'une activité définie au niveau supérieur. Dans le cas d'activités collaboratives, chaque couloir du diagramme d'activité permet de déterminer les tâches spécifiques conduites par un acteur ainsi que le séquençement de ces tâches. À chacun des niveaux d'abstraction, des diagrammes de classes et des diagrammes états-transitions permettent de caractériser/préciser les éléments de modélisation mis en évidence dans les autres diagrammes.

Par leur capacité à décrire un scénario à différents niveaux d'abstraction, PoEML (*cf.* figure 11) comme CPM et coUML permettent de mettre en relation des modèles du scénario avec des représentations abstraites des fonc-

tionnalités d'outils ou de services censées supporter un tel scénario. *Ces langages visuels offrent ainsi des moyens de médiation entre les pratiques pédagogiques et les infrastructures informatiques actuelles* (Motschnig-Pitrik et Derntl, 2005). Cette médiation peut conduire les concepteurs à repenser certains aspects du scénario imaginé parce qu'il n'est pas possible de rendre opérationnel un tel scénario sur une infrastructure donnée. Lorsque l'infrastructure d'exécution le permet, cette médiation peut aussi conduire les concepteurs à spécialiser certaines fonctionnalités d'outils offertes par l'infrastructure choisie pour que ces fonctionnalités respectent les attentes exprimées dans le scénario spécifié.

## Discussion

Dans un article récent, Botturi, Derntl, Boot et Gigl (2006) proposent une taxonomie des langages dits de modélisation pédagogique. Selon cette taxonomie, tout langage peut se caractériser selon son niveau de stratification (un ou plusieurs niveaux d'abstraction), son niveau de formalisation (de l'informel au formel), son niveau d'élaboration (du conceptuel à l'implémentation), le système de notation qu'il offre



aux concepteurs (textuel ou visuel) et le nombre de perspectives (une ou plusieurs) qu'il propose pour analyser une situation pédagogique.

Notre étude nous semble complémentaire de celle de Botturi, Derntl *et al.* (2006) dans la mesure où nous avons établi qu'au-delà des différences entre langages que révèle la taxonomie, il existe plusieurs convergences notables entre les langages visuels. Il faut souligner tout d'abord le fait que les langages étudiés essaient de mettre les concepteurs en situation authentique en proposant des primitives proches de leurs préoccupations pédagogiques quoique parfois éloignées des métaphores et primitives véhiculées par les infrastructures informatiques. Ces primitives de conception qui ne sont pas étudiées dans la taxonomie de Botturi, Derntl *et al.* (2006) sont pourtant pertinentes pour analyser les pers-

pectives de conception offertes par les langages visuels. Chaque perspective s'exprime alors avec un langage dédié ou DSL (*domain-specific language*) permettant de manipuler concrètement ces primitives de conception.

Il faut également prendre en compte le fait que les différents critères de la taxonomie ne sont pas considérés de manière indépendante par les équipes de recherche qui ont développé ces langages. C'est le cas des dimensions *perspective* et *stratification*, chaque perspective pouvant s'analyser à différents niveaux d'abstraction grâce à des primitives dédiées, conformément aux résultats présentés dans Gibbons et Stubbs (2007) selon lesquels il n'y a pas un langage de conception universel, mais un ensemble de langages pour aborder les problèmes et les sous-problèmes.

La figure 1. ci-dessous précise les relations entre perspective de modélisation et niveau de stratification :

Pour un niveau de stratification donné, un modèle du scénario jugé cohérent par une équipe de concepteurs est constitué d'un ensemble de représentations visuelles. Chacune de ces représentations est produite à l'aide d'une notation particulière (ou DSL, pour *domain specific language*) adaptée à la perspective choisie. Le nuage à la figure 1 signifie que cet ensemble de représentations constitue un tout cohérent du point de vue des concepteurs;

Les représentations produites à un niveau de stratification donné peuvent être réétudiées aux niveaux de stratification suivants (au niveau implémentation, il s'agit de confronter/adapter les spécifications du scénario décrit aux composants ou services offerts par l'infrastructure informatique sur laquelle ce scénario sera déployé);

Quand on change de niveau de stratification, l'étude d'une représentation visuelle peut conduire les concepteurs à produire plusieurs représentations visuelles complémentaires; ou à agréger les informations provenant de plusieurs représentations en une seule (ex. : plusieurs diagrammes de séquences agrégés en un diagramme état-transitions; un diagramme état-transitions qui vient préciser et compléter un diagramme de classes).

Ces éléments nous semblent importants à considérer pour qu'un langage visuel de modélisation atteigne les deux objectifs discutés dans cet article :

1. Supporter la créativité des concepteurs : dans ce cas, le langage est *un outil exploratoire* (cf. section 2.1 et les strates de niveau Conceptuel de la figure 1) utilisé pour décrire le problème de conception à résoudre et les grandes lignes du scénario pédagogique que les concepteurs envisagent de mettre en œuvre;
2. Rendre possible les échanges au sein d'une équipe de conception : le langage devient alors *un outil pour approfondir les modèles visuels issus de la phase*

*précédente*, pour les conforter et assurer la nécessaire médiation entre les processus pédagogiques exprimés et les processus que sont capables de mettre en œuvre les infrastructures informatiques actuelles (cf. section 2.2 et les strates de niveaux Spécification et Implémentation de la figure 1).

À l'issue de cette étude, il nous semble important de noter que la très grande majorité des langages visuels étudiés ne fournissent qu'un support méthodologique limité : les auteurs de ces langages visuels diffusent des études de cas montrant les modèles qu'ils ont produits, mais n'explicitent pas souvent la manière de procéder pour aboutir à de tels modèles. Bien que les relations entre perspectives de modélisation soient assez souvent formalisées dans ces études de cas, force est de constater que les environnements de modélisation permettant d'utiliser ces langages visuels :

- n'aident guère les concepteurs à mettre en relation les différentes représentations du scénario pédagogique qu'ils produisent à un niveau de stratification donné (cf. l'axe horizontal de la figure 1). La plupart de ces environnements de modélisation gagneraient donc à être accompagnés de guides de bonnes pratiques ainsi que de fonctionnalités capables de détecter les incohérences entre différentes représentations visuelles d'un scénario;
- n'incitent guère les concepteurs à mettre en relation les différentes strates de représentations visuelles produites pour un scénario pédagogique (cf. l'axe vertical de la figure 1). La nécessaire discussion qui doit s'instaurer au sein d'une équipe pluridisciplinaire de concepteurs (Baker, 2006) dépend pourtant de la capacité de cette équipe à mettre en relation (et confronter) les différentes strates de représentations visuelles produites. Il semble donc important de doter les environnements de modélisation de fonctionnalités facilitant cette confrontation.

À ce jour, le support méthodologique associé aux langages visuels étudiés nous paraît donc insuffisant pour permettre à une majorité d'enseignants de se saisir de ces langages dans leurs pratiques de scénarisation pédagogique. Considérant que les problèmes se situent sur le plan tant de la sémantique

que de la syntaxe de ces langages visuels, ce dernier point nous incite à approfondir les travaux de recherche pour mettre en cohérence, quel que soit le niveau d'abstraction considéré, les primitives de conception et les perspectives de modélisation utilisées pour décrire un EIAH.

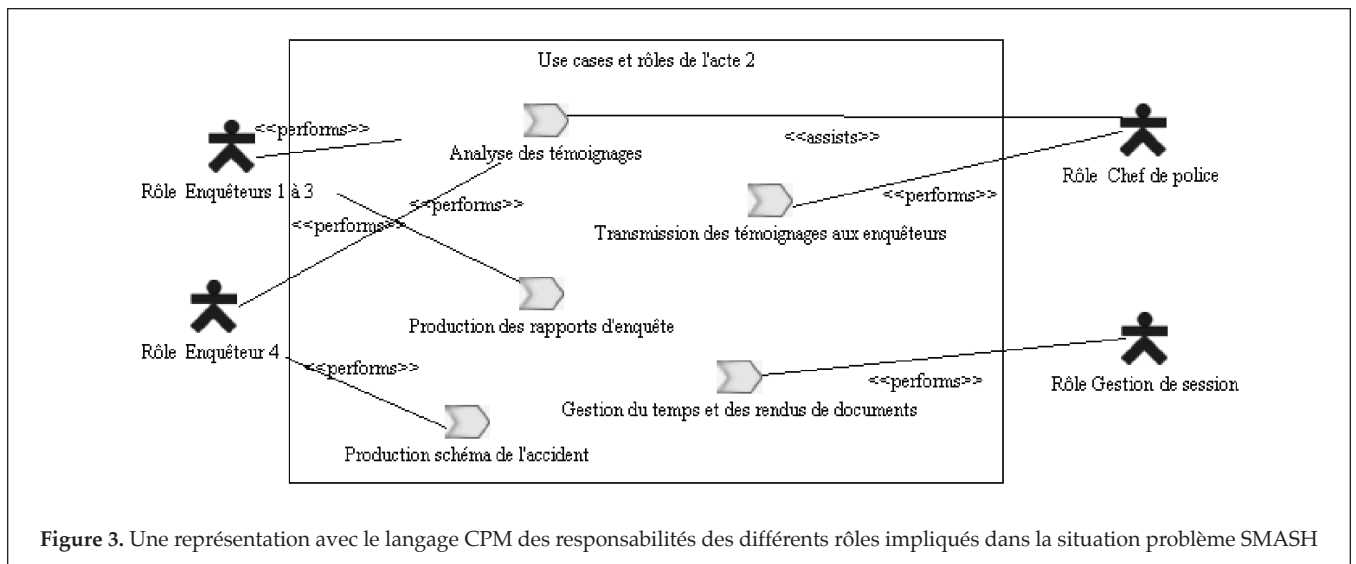
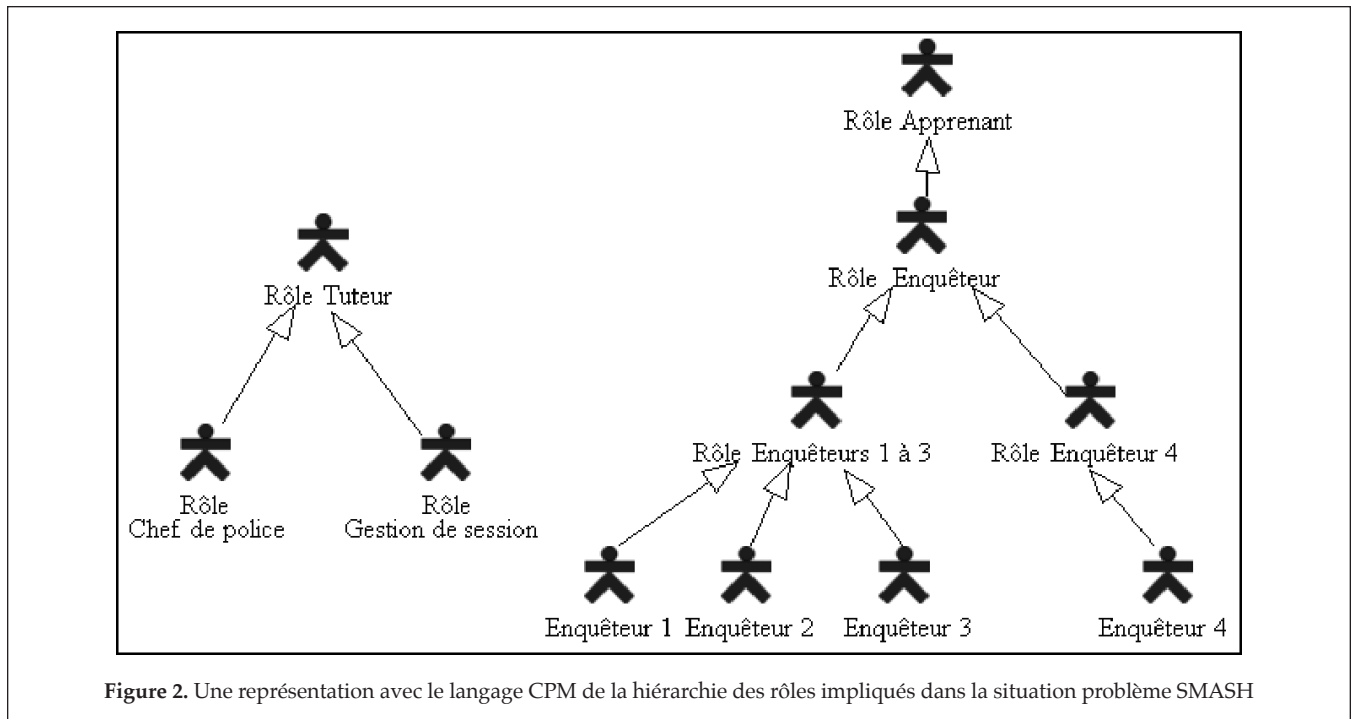
## Références

- Anderson, L. W. et Krathwohl, D. R. (2001). *A taxonomy for learning, teaching and assessing. A revision of Bloom's taxonomy of educational objectives*. New York : Addison Wesley Longman.
- Baker, M. (2006, juin). *Rôle et usage des modèles en environnements informatiques pour l'apprentissage humain*. Communication présentée au Colloque Ingénierie des Connaissances (IC'2006), organisé dans le cadre de la Semaine de la Connaissance, Nantes, France.
- Botturi, L. (2003). *E<sup>3</sup>ML: Educational environment modeling language*. Thèse de doctorat non publiée, Université de Lugano, Italie.
- Botturi, L., Cantoni, L., Lepori, B. et Tardini, S. (2006). Fast prototyping as a communication catalyst for e-learning design. Dans M. Bullen et D. Janes (dir.), *Making the transition to e-learning: Strategies and issues* (p. 266-283). Hershey, PA : Idea Group.
- Botturi, L., Derntl, M., Boot, E. et Gigl, K. (2006, juillet). *A classification framework for educational modeling languages in instructional design*. Communication présentée à la Sixth IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade, Pays-Bas.
- Burnett, M. (mise à jour 2007). *Visual language research bibliography*. Récupéré le 26 avril 2007 du site de l'auteure au College of Engineering de l'Oregon State University : <http://web.engr.oregonstate.edu/~burnett/vpl.html>
- Caeiro-Rodriguez, L., Llamas-Nistal, M. et Anido-Rifon, L. (2006, septembre). *The PoEML proposal to model services in educational modeling languages*. Communication présentée au International Workshop on Groupware (CRIWG), Medina del Campo, Espagne.
- Cook, S. (2002, mars). *Model-driven approaches to software development*. Communication présentée au colloque Objets, Composants, Modèles (OCM 2002) : l'évolution des architectures logicielles et des méthodes de développement, Nantes, France.
- Cross, N. (2001). Design cognition: Results from protocol and other empirical studies of design activity. Dans C. M. Eastman, W. M. McCracken et W. C. Newstetter (dir.), *Design knowing and learning: Cognition in design education* (p. 79-103). Amsterdam : Elsevier.
- Derntl, M. (2005). *Patterns for person-centered e-learning*. Thèse de doctorat non publiée, Université de Vienne, Autriche.
- Derntl, M. et Motschnig-Pitrik, R. (2007). coUML – A visual modeling language for cooperative environments. Dans L. Botturi et S. T. Stubbs (dir.), *Handbook of visual languages in instructional design: Theories and practices* (p. 155-184). Hershey, PA : IDEA Group.
- Dillenbourg, P. et Tchounikine, P. (2007). Flexibility in macro-scripts for computer-supported collaborative learning. *Journal of Computer Assisted Learning*, 23(1), 1-13.
- Ferraris, C. et Martel, C. (2007). LDL for collaborative activities. Dans L. Botturi et T. Stubbs (dir.), *Handbook of visual languages for instructional design: Theories and practices* (p. 226-253). Hershey, PA : IDEA Group.
- Ferruci, F., Tortora, G. et Vitello, G. (2002). Exploiting visual languages in software engineering. Dans Chang S. K. (dir.), *Handbook of software engineering and knowledge engineering*. River Edge, NJ : Singapore World Scientific.
- Fowler, M. (2003). *UML distilled: A brief guide to the Standard Object Modeling Language* (3<sup>e</sup> éd.). Boston : Addison-Wesley.
- Gagné, R. M., Briggs, L. J. et Wager, W. W. (2005). *Principles of instructional design* (5<sup>e</sup> éd.). New York : Holt, Rinehart and Winston.
- Gibbons, A. (2003). What and how do designers design? A theory of design structure. *TechTrends*, 47(5), 22-27.
- Gibbons, A. et Stubbs, S. T. (2007). The Pervasiveness of Design Drawing in ID. Dans L. Botturi et S. T. Stubbs (dir.), *Handbook of visual languages in instructional design: Theories and practices* (p. 345-365). Hershey, PA : IDEA Group.
- Griffiths, D. et Blat, J. (2005). The role of teachers in editing and authoring units of learning using IMS learning design. *Advanced Technology for Learning*, 2(4), article 208-0867.
- Gronlund, N. E. (1995). *How to write and use instructional objectives* (5<sup>e</sup> éd.). Englewood Cliffs, NJ : Prentice Hall.
- Hernández-Leo, D., Villasclaras-Fernández, E. D., Asensio-Pérez, J. I., Dimitriadis, Y., Jorrín-Abellán, I. M., Ruiz-Requies, I. et al. (2006). COLLAGE: A collaborative learning design editor based on patterns. *Educational Technology & Society*, 9(1), 58-71.
- Holoyak, K. J. (1991). *Symbolic connectionism: Toward third generation-theories of expertise*. New York : Cambridge University Press.
- IMS Global Learning Consortium. (2003a). *IMS learning design best practice and implementation guide*. Lake Mary, FL : auteur.
- IMS Global Learning Consortium. (2003b). *IMS learning design information model*. Lake Mary, FL : auteur.

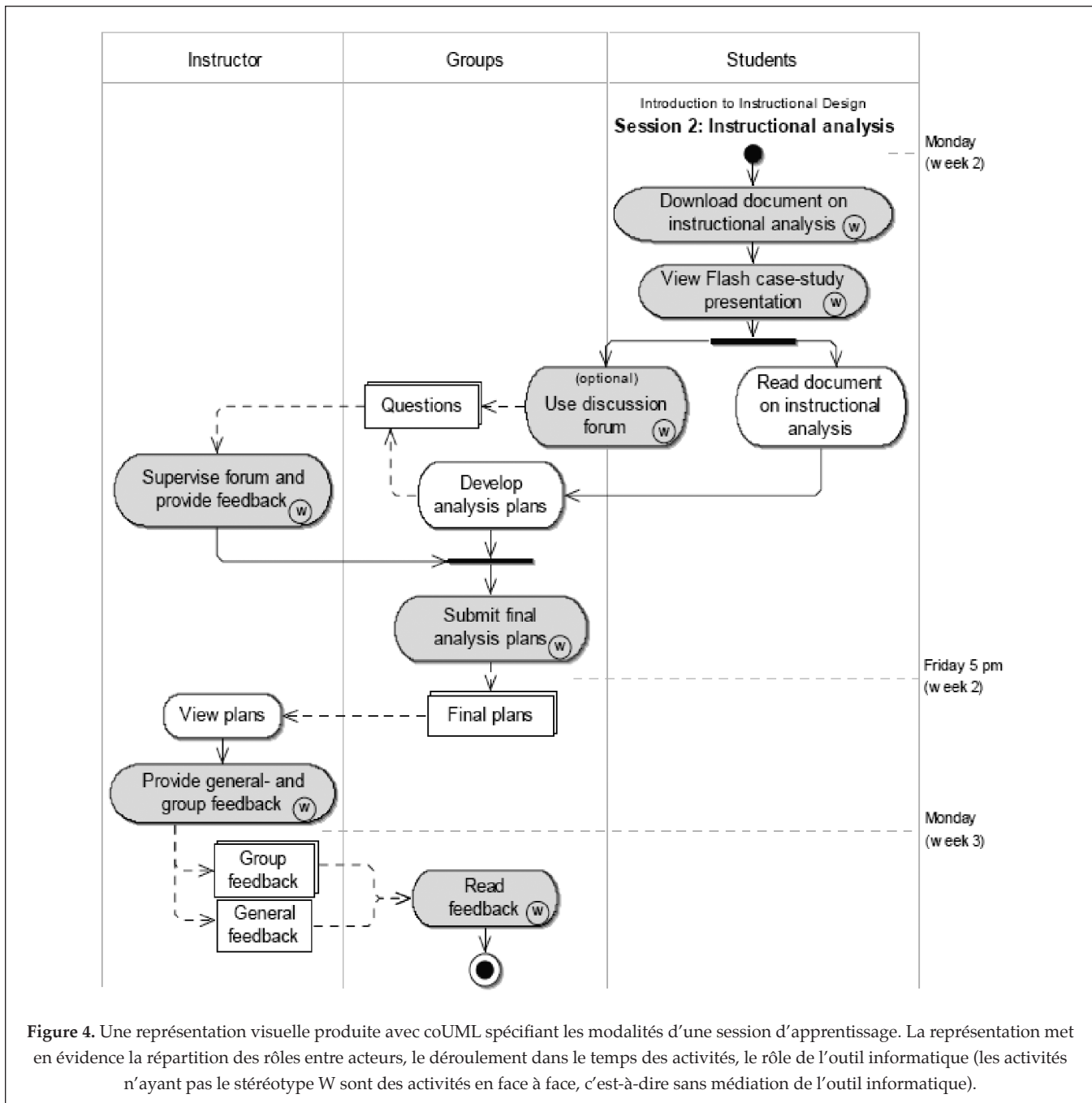
- 
- Laforcade, P., Nodenot, T. et Sallaberry, C. (2005). Un langage de modélisation pédagogique basé sur UML. Dans S. George et A. Derycke (dir.), *Conceptions et usages des plates-formes de formation* [numéro spécial]. *Sciences et technologies de l'information et de la communication pour l'éducation et la formation*, 12.
- Marriott, K. et Meyer, B. (1998). *Visual language theory*. New York : Springer-Verlag.
- Merrill, M. D. (1994). *Instructional design theory*. Englewood Cliffs, NJ : Educational Technology.
- Motschnig-Pitrik, R. et Derntl, M. (2005, octobre). *Learning process models as mediators between didactical practice and Web support*. Communication présentée à la International Conference on Conceptual Modeling (ER'2005), Klagenfurt, Autriche.
- Nodenot, T., Laforcade, P. et Le Pallec, X. (2007). Visual design of coherent technology-enhanced learning systems: A few lessons learnt from CPM language. Dans L. Botturi et S. T. Stubbs (dir.), *Handbook of visual languages in instructional design: Theories and practices* (p. 254-280). Hershey, PA : IDEA Group.
- Paquette, G., Léonard, M., Lundgren-Cayrol, K., Mihaila, S. et Gareau, D. (2006). Learning design based on graphical knowledge-modeling. *Journal of Educational Technology & Society*, 9(1), 97-112.
- Sampson, D., Karampiperis, P. et Zervas, P. (2005). ASK-LDT: A Web-based learning scenarios authoring environment based on IMS learning design. *Advanced Technology for Learning*, 2(4), 207-215.
- Stubbs, S. T. et Gibbons, A. (2007). The power of design drawings in other design fields. Dans L. Botturi et T. Stubbs (dir.), *Handbook of visual languages in instructional design: Theories and practices* (p. 33-51). Hershey, PA : IDEA Group.
- Verpoorten, D., Poumay, M. et Leclercq, D. (2006, mars). *The 8 learning events model: A pedagogic conceptual tool supporting diversification of learning methods*. Communication présentée au International Workshop in Learning Networks for Lifelong Competence Development – TENCompetence Conference, Sofia, Bulgarie.
- Villiot-Leclercq, E. et David, J.-P. (2007). Le formalisme des Pléiades pour la conception et l'adaptation de patrons de scénarios pédagogiques. Dans T. Nodenot, J. Wallet et E. Fernandes (dir.), *Actes de la conférence Environnement informatique pour l'apprentissage humain (EIAH 2007)* (p. 269-274). Paris/Lyon : Association des technologies de l'information pour l'éducation et la formation [ATIEF] et Institut national de recherche pédagogique [INRP].

## Annexe

### Des exemples de représentations typiques des langages visuels dédiés à la conception d'EIAH







**Figure 4.** Une représentation visuelle produite avec coUML spécifiant les modalités d’une session d’apprentissage. La représentation met en évidence la répartition des rôles entre acteurs, le déroulement dans le temps des activités, le rôle de l’outil informatique (les activités n’ayant pas le stéréotype W sont des activités en face à face, c’est-à-dire sans médiation de l’outil informatique).

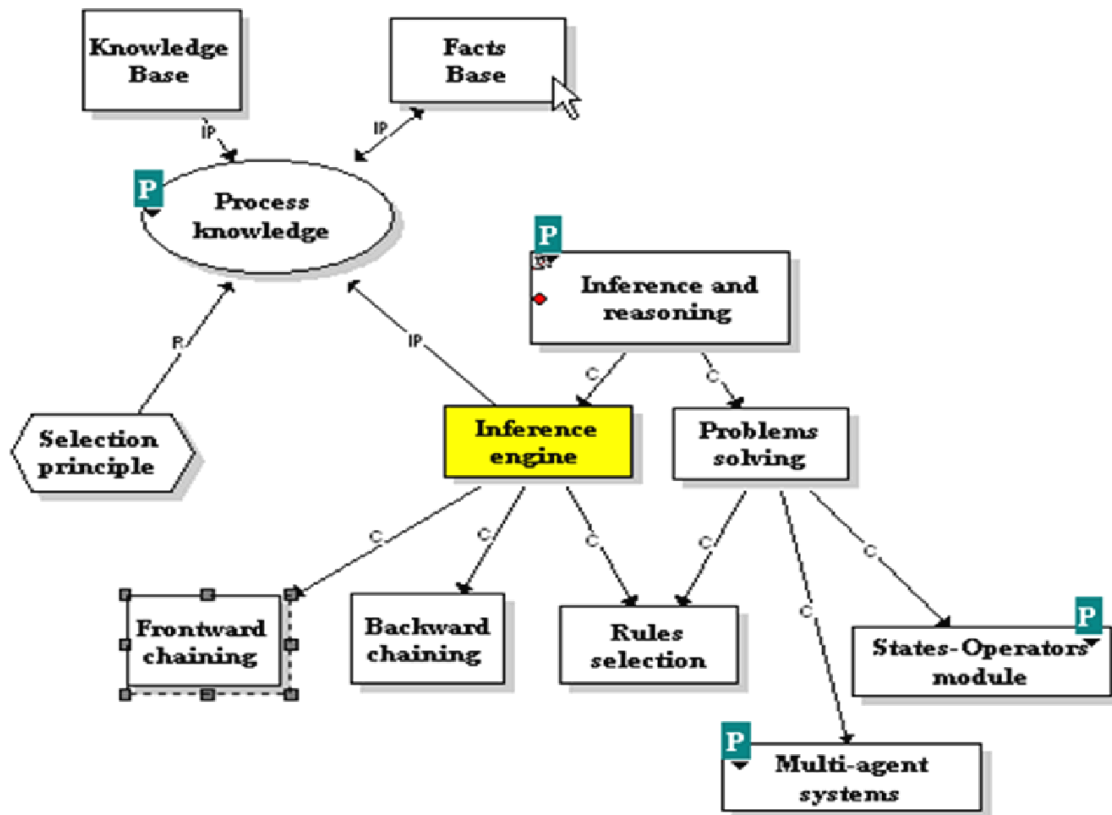


Figure 5. Une représentation visuelle produite avec MOT+ pour décrire le domaine enseigné dans le cadre des activités d'apprentissage met en évidence la répartition des rôles entre acteurs, le déroulement dans le temps des activités, le rôle de l'outil informatique (les activités n'ayant pas le stéréotype W sont des activités en face à face, c'est-à-dire sans médiation de l'outil informatique).

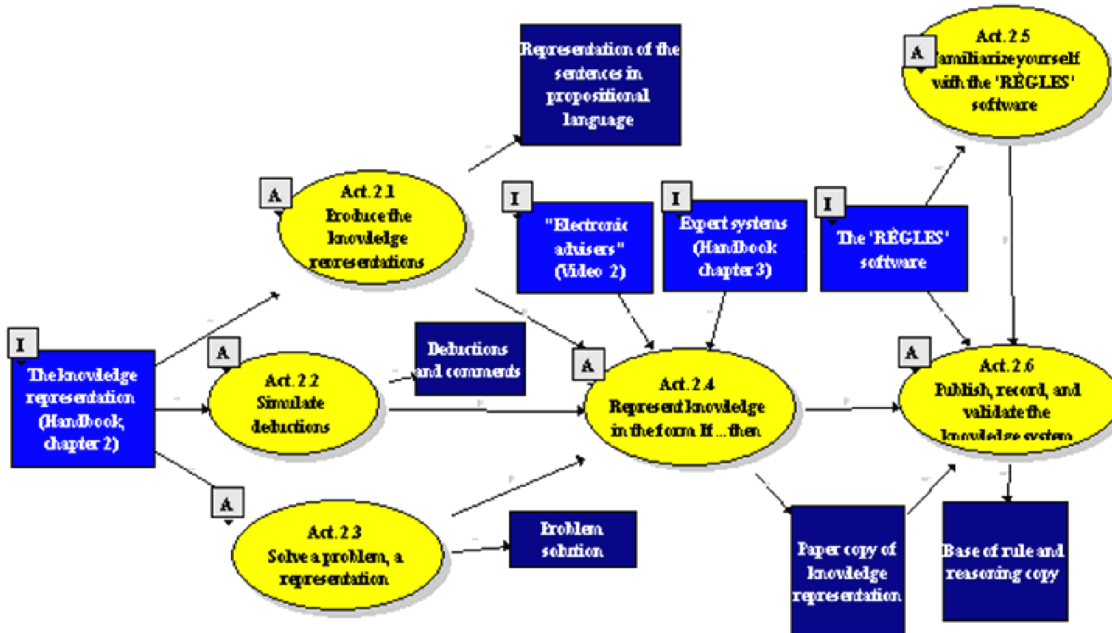


Figure 6. Une représentation visuelle produite avec MOT+ pour lier le domaine enseigné (cf. figure 5) au modèle de tutorat

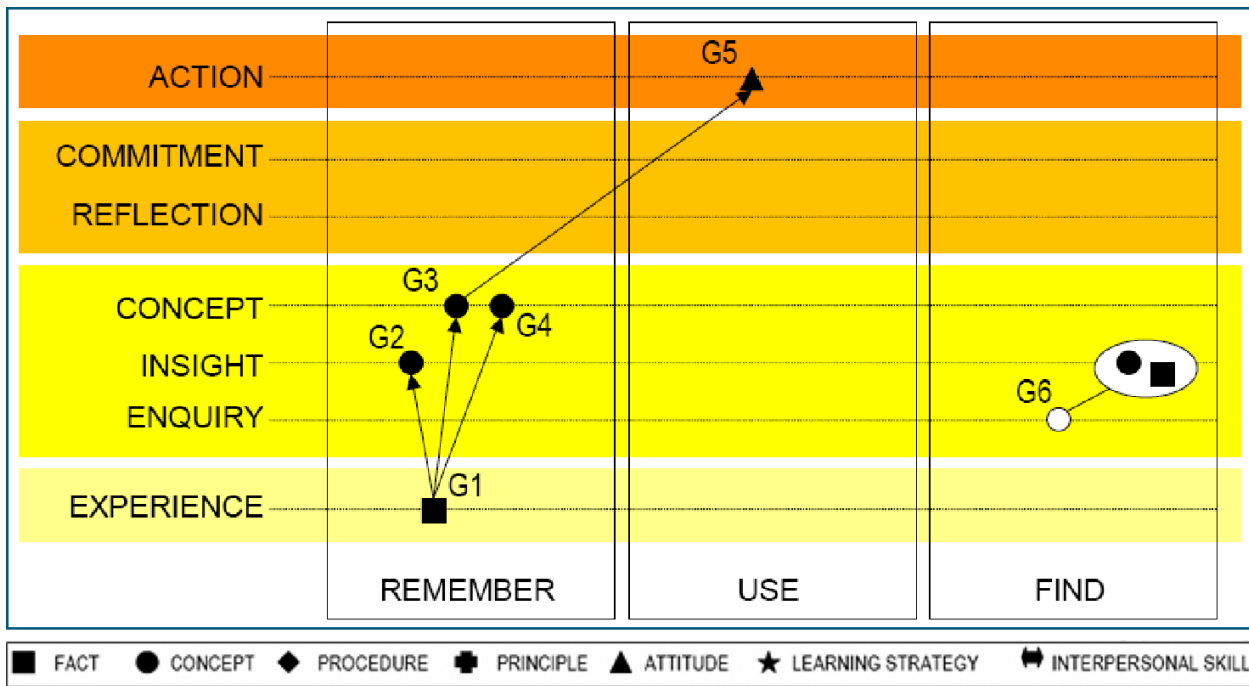


Figure 7. Une représentation visuelle produite avec E<sup>2</sup>ML positionnant les objectifs d'apprentissage d'une session. On y remarque la typologie des objectifs d'apprentissage (fait, concept, procédure, principe, attitude, etc.), un degré de perception de ces objectifs en quatre niveaux (expérience, compréhension, prise de recul, disponibilité) et une métrique identifiant la façon dont l'apprenant peut mobiliser ces objectifs (simple mémorisation, utilisation dans des situations spécifiques, exploitation pour générer de nouvelles connaissances).

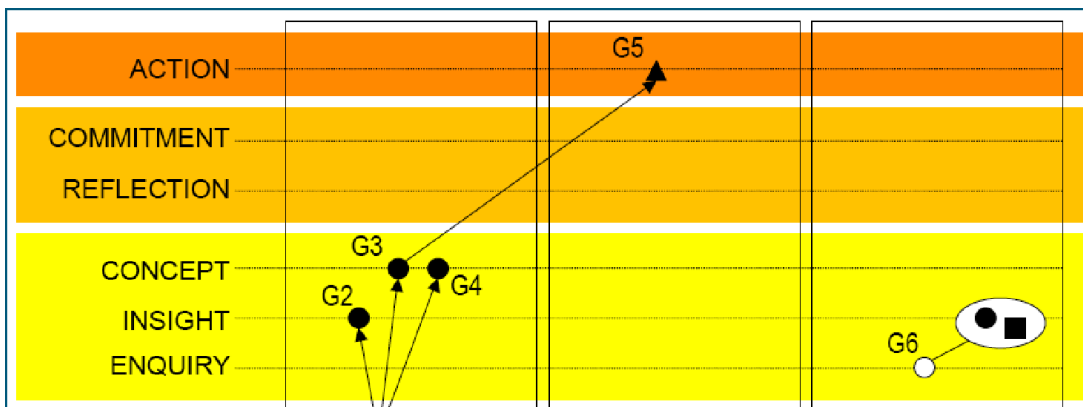
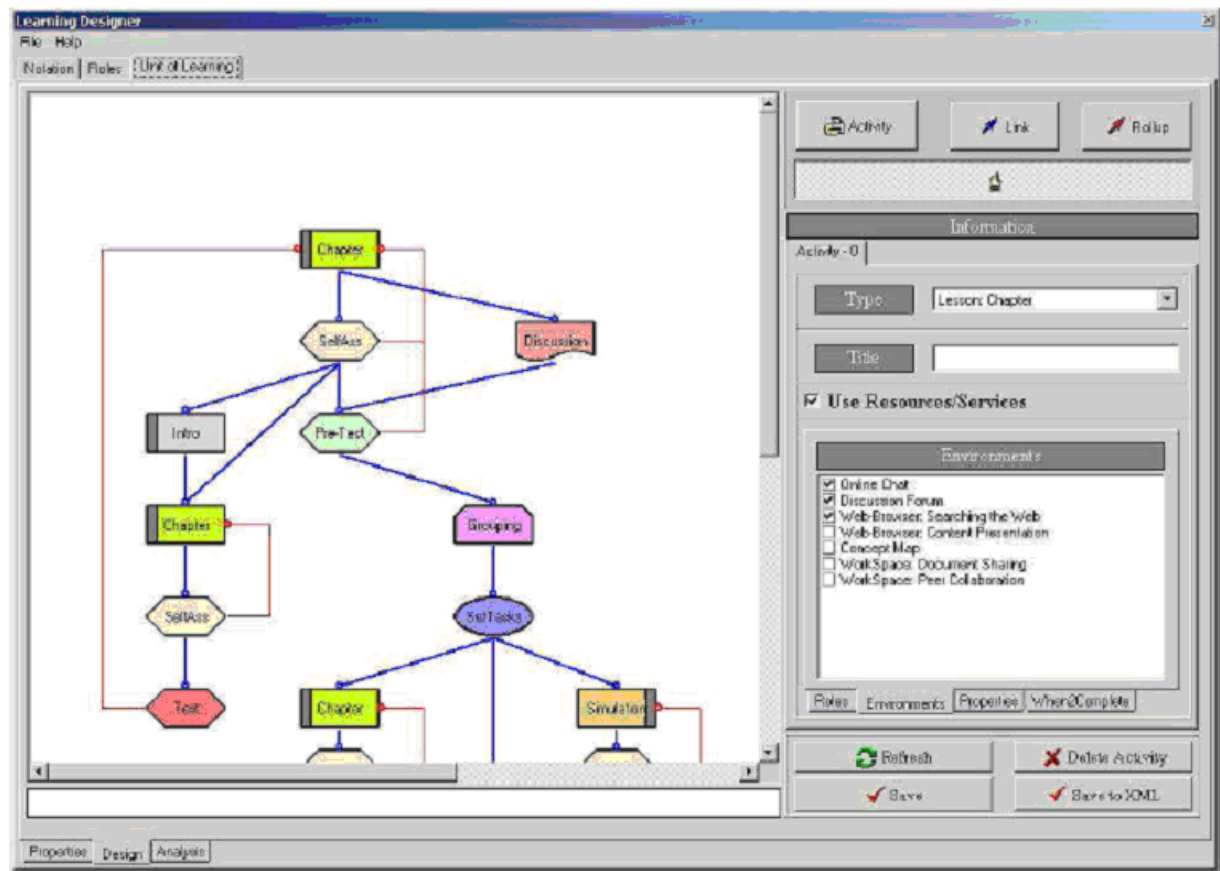


Figure 8. Un patron de représentation visuelle pour décrire avec E<sup>2</sup>ML la manière d'atteindre les objectifs d'apprentissage identifiés



**Figure 9.** Une représentation visuelle produite avec ASK-LDT pour représenter la structuration d'un module d'enseignement en activités d'apprentissage. Basé sur la spécification IMS-LD, l'éditeur permet de structurer les activités d'une session d'apprentissage, d'associer ces activités à des rôles d'acteurs ET de définir le flux de contrôle de ces activités sur la base de conditions portant sur les valeurs prises par des propriétés (cf. niveaux B et C de la spécification IMS-LD).

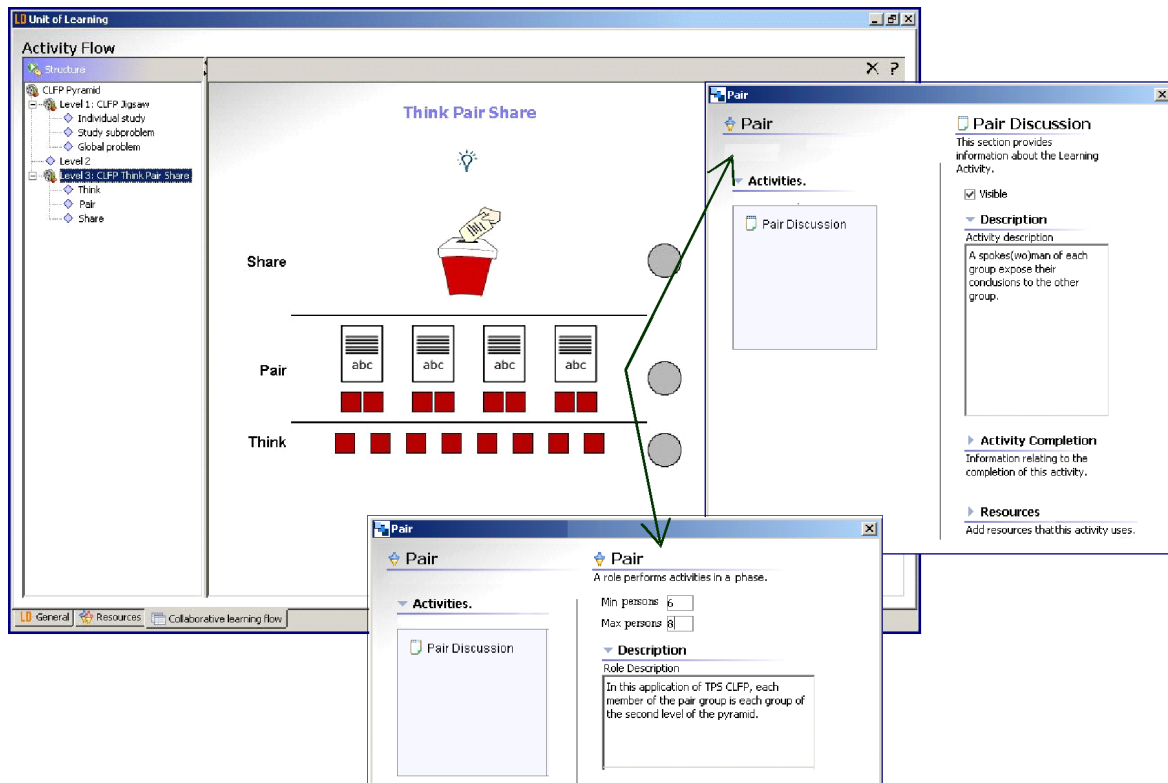


Figure 10. Une représentation visuelle (patron) proposée par COLLAGE pour décrire les collaborations entre acteurs (ici le patron Think – Pair – Share)

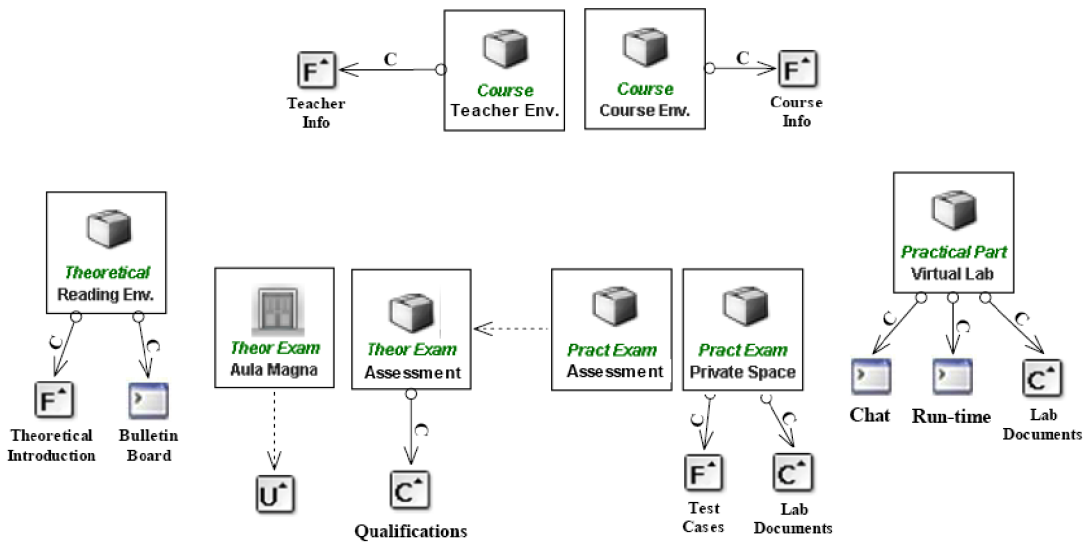


Figure 11. Une représentation visuelle produite avec PoEML pour associer chacune des unités d'enseignement à un environnement spécifique de mise en œuvre