



UNIVERSITÉ
DE GENÈVE

FACULTÉ DE PSYCHOLOGIE
ET DES SCIENCES DE L'ÉDUCATION



BPMN 2.0 POUR LA MODÉLISATION ET L'IMPLÉMENTATION DE DISPOSITIFS PÉDAGOGIQUES ORIENTÉS PROCESSUS.

Julien Da Costa

Mémoire présenté pour l'obtention du Master MALTT
Master of Science in Learning and Teaching Technologies
Unité TECFA
Faculté de psychologie et des sciences de l'éducation

GENÈVE, FÉVRIER 2014

Jury

Daniel K. Schneider	Maître d'enseignement et de recherche, TECFA	Directeur
Nicolas Szilas	Maître d'enseignement et de recherche, TECFA	Examinateur
Alain Buchet	Formateur indépendant Ex-chef d'unité Learning Solution, CICR	Examinateur

RÉSUMÉ

Dans le cadre de ce mémoire nous avons exploré la possibilité d'utiliser BPMN 2.0 en tant que langage graphique de modélisation pédagogique. En étudiant une technologie issue de l'ingénierie des processus métier, notre démarche contribue à ouvrir l'horizon des technologies éducatives et vise plus globalement à soutenir une réflexion sur l'avenir des environnements numériques d'apprentissage et la place des intentions pédagogiques dans les processus de conception. Nous proposons une piste pour le développement de dispositifs d'apprentissage basés sur les scénarios en utilisant les cadres conceptuels de la gestion des processus métier (BPM), de l'ingénierie dirigée par les modèles (IDM) et de l'architecture orientée services (SOA). L'enjeu est de savoir en quoi la logique de « workflow » propre à ce langage est compatible avec celle d'un scénario pédagogique et ce que pourrait apporter l'utilisation d'un tel standard à l'ingénierie pédagogique. Notre exploration nous a amené à proposer une palette d'éléments graphiques pour la modélisation pédagogique avec BPMN et à utiliser des outils de développement spécialisés dans la conception d'infrastructures BPM afin de tester leur implémentation.

TABLE DES MATIÈRES

INTRODUCTION.....	1
1. Motivation de la recherche et problématique.....	1
2. Structure du mémoire.....	3
CHAPITRE I. DES THÉORIES ÉDUCATIVES À LA SCENARISATION PÉDAGOGIQUE	4
Introduction	4
1. Modèle bémoriste.....	5
1.1 Origine d'une approche comportementale	5
1.2 Adaptation à l'apprentissage humain.....	5
1.3 Un héritage important malgré de nombreuses critiques.....	6
2. Modèle constructiviste.....	7
2.1 Approche selon l'épistémologie génétique	7
2.2 Approche historico-culturelle.....	7
2.3 Apports du constructivisme à la pédagogie moderne.	8
2.4 TICE et constructivisme.....	8
3. Modèle cognitiviste	9
3.1 Aux origines du traitement cérébral de l'information.....	9
3.2 Apports à l'apprentissage	10
3.3 TICE et cognitivisme	10
4. Modèle socioconstructiviste.....	11
4.1 Approche européenne – Évolution de l'interactionnisme Piagétien.	11
4.2 Approche américaine – Poursuite des travaux de l'école russe.....	12
4.3 TICE et socioconstructivisme.....	13
Conclusion.....	14
CHAPITRE II. SCÉNARISATION DE L'ENSEIGNEMENT	15
Introduction	15
1. De l'instructional design à la scénarisation pédagogique.....	15
1.1 Définition de l>ID.....	15
1.2 Définition de la scénarisation pédagogique.....	16
2. Théorie de l'activité un cadre conceptuel.....	18
2.1 L'activité une entité composite.....	18
2.2 Représentations triangulaire de l'activité.....	19
2.3 TA et scénarisation pédagogique.	20
Conclusion.....	21
CHAPITRE III. INGÉNIERIE PÉDAGOGIQUE	22
Introduction	22

1. Standardiser les objets d'apprentissage	22
1.1 D'une centration sur le contenu, à une centration sur l'activité.	22
1.2 - IMS CP, LOM, SCORM.	23
1.3 – EML, IMS Learning Design.	24
2. Script CSCL	25
2.1 Computer-Supported Collaborative Learning.....	25
2.2 Notion de script.....	26
3. Langage de modélisation pédagogique.....	27
3.1 Définition d'un LMP.	27
3.2 Panorama des recherches en LMP.....	28
3.3 Synthèse.	35
4. Ingénieur pédagogique ?	35
4.1 Artisanat et ingénierie.....	35
4.2 Dispositifs instrumentés et transformations des acteurs.....	36
5. Ingénierie pédagogique et ingénierie des EIAH	38
5.1 Ingénierie dirigée par les modèles.....	39
5.2 Architecture orientée services.....	41
5.3 Enjeux pour les EIAH.....	43
Conclusion.....	45

CHAPITRE IV. BPMN 2.0 - INTRODUCTION AU STANDARD 46

Introduction	47
1. De l'approche BPM au standard BPMN 2.0.....	47
1.1 Ingénierie des processus métier (BPM).	47
1.2 BPMN – un standard international pour l'ingénierie des processus.....	49
1.3 De BPMN 1.2 à BPMN 2.0.....	49
2. BPMN 2.0.....	50
3. Outils et principes de développement.....	52
3.1 Outils de modélisation.	53
3.2 Outil de développement complet (BPMS).	54
Conclusion.....	56

CHAPITRE V. MODÉLISATION DE SCÉNARIO PÉDAGOGIQUE AVEC BPMN 57

Introduction	57
1. BPMN sous un paradigme éducatif.....	58
1.1 BPMN en tant que Langage graphique de modélisation pédagogique.....	58
1.2 Cycle de vie d'un scénario pédagogique et approche BPM.	60
1.3 Analyse des traces de processus et analyse de données de formation.	61
2. Proposition d'utilisation.....	62
2.1 Palette d'éléments.	62
2.2 Cadre de conception.....	65

2.3 Une plateforme pour l'exécution de scénarios.....	68
3. Étude de cas	70
3.1 Activité simple de correction	70
3.2 Écriture de récits collaboratifs en contexte hybride.....	72
Conclusion et discussion	77
CONCLUSION ET PERSPECTIVES	79
Résumé et bilan.....	79
Défis à relever pour la mise en œuvre de BPMN en éducation	80
Enjeux et limites d'une approche par workflow	81
BIBLIOGRAPHIE	84
ANNEXES.....	92
Annexe 1 - historique des courants de l'apprentissage	92
Annexe 2 – Complément IMS LD.....	93
A2-1 Définition d'une unité d'apprentissage et modèle structurel	93
A2-2 Niveaux de conception et modèle relationnel :	94
Annexe 3 – PoEML perspectives.....	95
Annexe 4 - Complément au chapitre 4 - BPMN 2.0.....	96
A4-1. Structure du langage BPMN.....	96
A4-1.1 Sous-classes de conformité.....	96
A4-1.2 Meta-modèles.....	96
A4-2. Les types de symboles BPMN	98
A4-2.1 Élément de flux	98
A4-2.2 Éléments de connexion	99
A4-2.3 Bassins et couloirs	99
A4-2.4 Artefacts	101
A4-3. Principes de modélisation.....	102
A4-3.1 Les différents type de diagrammes.....	102
A4-3.2 Décomposition hiérarchique, boucles et multi-instanciation	104
Annexe 5 – exemple wsdl 2.0.....	106
Annexe 6 – Quelques outils de modélisation gratuits.....	107
Annexe 7 – BPMS proposant une solution gratuite.....	108
Annexe 8 – capture d'écran de Bonita BPM 6.....	109
A7-1 – Accueil (type écosystème).....	109
A7-2 – Aperçu de l'interface de développement (définition de variable).....	109
A7-3 Aperçu création d'interface formulaire.....	110
Annexe 9 – Scénario BPMN récit collaboratif.....	111
Annexe 10 – Détails procédure de développement	112

TABLE DES FIGURES

FIGURE 1: PROCESSUS D'ASSIMILATION ET ACCOMMODATION SELON PIAGET (CRAHAY, 1999, P. 179).....	7
FIGURE 2: PRINCIPE SIMPLE DU TRAITEMENT DE L'INFORMATION SELON ELLEN D. GAGNÉ (1985).....	9
FIGURE 3: DÉVELOPPEMENT SOCIOCOGNITIF EN SPIRALE (DOISE, 1993).....	11
FIGURE 4: BASE DU SYSTÈME D'ACTIVITÉ POUR UNE PRATIQUE FLC. (BROWN & CAMPIONE, 1996, P. 293)	13
FIGURE 5: REPRÉSENTATION SCHÉMATIQUE DE LA SCÉNARISATION PÉDAGOGIQUE.....	17
FIGURE 6: EXEMPLE D'ACTIVITÉ, ACTION ET OPÉRATIONS SELON KUUTTI (2005)	19
FIGURE 7: STRUCTURE DE BASE DE L'ACTIVITÉ HUMAINE MÉDIATISÉE (NARDI, 1996, P.25).	19
FIGURE 8: STRUCTURE D'UNE ACTIVITÉ D'APRÈS ENGESTRÖM (1987)	20
FIGURE 9: ARCHITECTURE DU PACKAGE INTERCHANGE FILE (PIF).....	23
FIGURE 10: COMPENDIUMLD ILLUSTRATION.	29
FIGURE 11: LES 4 TYPES DE DIAGRAMMES E2ML (BOTTURRI, 2003).....	29
FIGURE 12: ENVIRONNEMENT AUTEUR DE LAMS DANS UN NAVIGATEUR.	30
FIGURE 13: ILLUSTRATION DES DIAGRAMMES DE COUML (ADAPTÉ DEPUIS FIGL & AL. 2010).....	31
FIGURE 14: DIAGRAMME UML SIMPLIFIÉ	32
FIGURE 15: SCHÉMA ENSEIGNANT CONCEPTEUR SELON DELACOUR (2008).....	37
FIGURE 16: ACTEURS ET DOMAINES FONCTIONNELS D'UN DISPOSITIF DE FORMATION À DISTANCE (VANTROYS, 2005, P.6) ...	38
FIGURE 17: SCHÉMA RÉCAPITULATIF - LA SCÉNARISATION SOUS LE PARADIGME DES DISPOSITIFS INSTRUMENTÉS	39
FIGURE 18: NIVEAUX DE MODÉLISATION SELON L'OMG (BÉZIVIN, 2003)	40
FIGURE 19: REPRÉSENTATION SCHÉMATIQUE DE L'ARCHITECTURE D'UN SERVICE WEB SELON L'APPROCHE SOA	42
FIGURE 20: PARADOXE DE LA POLYVALENCE DES EIAH	43
FIGURE 21: CLASSES PRINCIPALES ONTOLOGIE DAML-S (GARNER, 2004)	44
FIGURE 22: UN DIAGRAMME BPMN PUREMENT ILLUSTRATIF, RÉALISÉ AVEC BONITA BPM.....	46
FIGURE 23: REPRÉSENTATION COMMUNE D'UN CYCLE DE VIE DES PROCÉDURES MÉTIER.....	48
FIGURE 24: CYCLE DE VIE DES PROCÉDURES MÉTIER (GARTNER 2005)	48
FIGURE 25: BPM - CYCLE DE VIE DES PROCESSUS MÉTIER (DEBAUCHE & MEGARD, 2004)	48
FIGURE 26: PROCESSUS D'EXPÉDITION DES COMMANDES AVEC BPMN 2.0 (OMG, 2013, P.170).....	50
FIGURE 27: ÉLÉMENTS PRINCIPAUX	50

FIGURE 28: EXEMPLE ILLUSTRANT LE FONCTIONNEMENT DE BPMN.....	51
FIGURE 29: CONNECTEUR GÉNÉRIQUE À UN SERVICE DE MESSAGERIE	55
FIGURE 30: COMPARAISON DE BPMN 2.0 À D'AUTRE LMP SELON CRITÈRES DE CLASSIFICATION DE NODENOT (2008).....	59
FIGURE 31: CYCLE DE VIE D'UN MODULE D'ENSEIGNEMENT SELON VANTROYS (2003)	60
FIGURE 32: CYCLE DE VIE D'UN SCÉNARIO PÉDAGOGIQUE SELON PERNIN (2007).....	60
FIGURE 33: EXEMPLE ILLUSTRATIF - GESTION TEMPORELLE EN BPMN	64
FIGURE 34 : EXEMPLE ILLUSTRATIF 2 : SYNCHRONIE EN BPMN.....	65
FIGURE 35: EXEMPLE ILLUSTRATIF 3 : RESSOURCES, DONNÉES, PRODUITS EN BPMN.....	65
FIGURE 36: PRINCIPE DE DÉCOMPOSITION. NIVEAU 1 ET 2.....	66
FIGURE 37: ÉCRAN DE CONNEXION À LA PLATEFORME E-FLOW.....	69
FIGURE 38 : PORTAIL E-FLOW EN MODE ADMINISTRATEUR- INSTALLATION D'UN SCÉNARIO	69
FIGURE 39: MODÈLE SIMPLE PROCESSUS DE CORRECTION	70
FIGURE 40: DIAGRAMME BPMN FINAL ACTIVITÉ DE CORRECTION.....	71
FIGURE 41: ACTIVITÉ DE CORRECTION DISPONIBLE SUR E-FLOW POUVANT ÊTRE DÉMARRÉ PAR UN ÉTUDIANT	71
FIGURE 42: NOUVELLE TÂCHE DISPONIBLE POUVANT ÊTRE RÉALISÉ PAR UN ASSISTANT.....	71
FIGURE 43: RÉALISATION DE LA TÂCHE PAR UN ASSISTANT	72
FIGURE 44: SCÉNARIO RÉCIT COLLABORATIF (NIVEAU 1) - GRAND FORMAT VOIR ANNEXE 9.....	73
FIGURE 45: DESCRIPTION ACTIVITÉ « PRODUCTION » (NIVEAU 2).....	74
FIGURE 46: DESCRIPTION ACTIVITÉ « CONSIGNE ET GROUPEMENT » (NIVEAU 2).....	75
FIGURE 47: PROCESSUS DE RÉGULATION ENSEIGNANT DÉCLENCHÉE VIA SIGNAL	75
FIGURE 48: APERÇUS DE L'INTERFACE D'E-FLOW POUR LA GESTION DE DONNÉES DE FORMATION	76
FIGURE 49: MINIER. P, (2003). ANCRAJE HISTORIQUE ET DÉVELOPPEMENT DES COURANTS DE PENSÉE DE L'APPRENTISSAGE. HTTP://WWWENS.UQAC.CA/~PMINIER/ACT1/GRAFH1.HTM.....	92
FIGURE 50: MODÈLE STRUCTUREL IMS-LD D'UNE UNITÉ D'APPRENTISSAGE. (ADAPTÉ DEPUIS IMS LD INFORMATION MODEL)	93
FIGURE 51: MODÈLE RELATIONNEL D'IMS-LD (ADAPTÉ DEPUIS IMS LD INFORMATION MODEL).....	94
FIGURE 52: MODÈLE UML DES 13 PERSPECTIVES DE POEML ET LEURS RELATIONS	95
FIGURE 53: MODÈLE UML DES ÉLÉMENTS PRINCIPAUX DE LA PERSPECTIVE "GOAL"	95
FIGURE 54: DIAGRAMME DE CLASSE "DEFINITIONS" (OMG, 2013, P.55).....	97
FIGURE 55: BPMN 2.0 STRUCTURE EN COUCHE EXTENSIBLE	97
FIGURE 56: ILLUSTRATION D'ÉLÉMENTS SIMPLES DANS UN PROCESSUS BPMN	99
FIGURE 57: ILLUSTRATION FONCTIONNEMENT DES ÉLÉMENTS "BASSIN" ET "COULOIR"	100

FIGURE 58: ILLUSTRATION DIAGRAMME DE COLLABORATION.....	100
FIGURE 59: ILLUSTRATION, PROCESSUS AVEC ARTEFACTS.....	101
FIGURE 60: EXEMPLE D'UN PROCESSUS PUBLIC (OMG, 2013, P.24) [TRADUIT EN FRANÇAIS ET COLORÉ].....	102
FIGURE 61: EXEMPLE DE DIAGRAMME DE CHORÉGRAPHIE (OMG, 2013, P.25) [TRADUIT]	103
FIGURE 62: REPRÉSENTATION SIMPLE D'UN DIAGRAMME DE CONVERSATION BPMN	104
FIGURE 63: OUVERTURE D'UN SOUS-PROCESSUS LORS D'UN CLIC SUR L'ÉLÉMENT.	104
FIGURE 64: PROCESSUS Ad-Hoc (OMG, 2013, P.182-183) [TRADUIT ET COLORÉ]	105
FIGURE 65: EXEMPLE ILLUSTRATIF D'UN DOCUMENT WSDL 2.0 (EXTRAIT WIKIPÉDIA)	106

LISTE DES TABLEAUX

TABLEAU 1: COMPARAISON DE BPMN 2.0 À D'AUTRE LMP SELON CRITÈRES DE CLASSIFICATION DE BOTTURI & AL. (2006).	59
TABLEAU 2 : PERSPECTIVE ET NIVEAU DE GRANULARITÉ LORS DE LA MODÉLISATION DE WORKFLOW BPMN	67
TABLEAU 3: COMPARAISON D'UN ÉLÉMENT BPMN SELON SOUS-CLASSE DE CONFORMITÉ	96
TABLEAU 4: ÉLÉMENT DE FLUX	98
TABLEAU 5: ÉLÉMENT DE CONNEXION.....	99
TABLEAU 6: ÉLÉMENTS BPMN « BASSIN » ET « COULOIR »	99
TABLEAU 7: ÉLÉMENTS ARTEFACTS	101

INTRODUCTION

Dans ce mémoire nous proposons une approche de modélisation et d'implémentation utilisant les cadres conceptuels de la gestion des processus métier (BPM¹), de l'ingénierie dirigée par les modèles (IDM) et de l'architecture orientée services (SOA²). Nous explorons la possibilité de modélisation d'activités pédagogiques via un langage graphique de description de workflow (BPMN), de manière à définir des scénarios riches orientés processus avec des possibilités d'implémentation et de connexion à des services externes.

Nous abordons ce travail dans le cadre d'une exploration d'une technologie évoquée dans la littérature comme une alternative possible aux langages de modélisation pédagogique mais jamais réellement investiguée. Notre suggestion sort du stricte cadre des technologies éducatives pour s'intéresser à ce qui est « en vogue » et fonctionne dans d'autres contextes afin d'en tirer de possibles applications pour l'éducation. Notre exploration nous a amené à déployer une plateforme de gestion de workflows à partir d'outils BPM spécialisés et tester l'exécution de scénario sur celle-ci.

1. MOTIVATION DE LA RECHERCHE ET PROBLÉMATIQUE

Les situations pédagogiques instrumentées par les TICE sont aujourd'hui incontournables, que ce soit dans le contexte des institutions ou dans celui des entreprises. Le choix, la mise en place, voir le développement de technologie de support à l'apprentissage fait partie intégrante de la mise en œuvre de dispositifs de formation qui intègrent de manière croissante les dimensions ligne/hors-ligne, distance/présence, individuel/collaboratif, formel/informel, théorie/pratique afin d'enrichir les modes de formation « traditionnels » et ce avec un rapport qualité prix raisonnable (Singh, 2003). En outre ces évolutions s'accompagnent d'une modification du rôle des acteurs et de la variété des tâches qui leur incombent (Bunt & al. 2001 ; Pernin 2003 ; Schneider & al. 2003). La dualité enseignant/apprenant est souvent trop réductrice pour aborder le développement des dispositifs de formation instrumentés. En institution comme en entreprise il est fréquent d'y ajouter le développement d'informaticien, le support d'une équipe IT, la gestion budgétaire d'un manager ou l'expertise d'un ingénieur pédagogique.

Parallèlement à ces évolutions, l'intérêt pour les pédagogies actives associées aux courants de pensées (socio)constructivistes et situationnistes amènent à repenser les dispositifs autour de l'apprenant avec une centration marquée, non pas sur les contenus, mais sur la manière dont ces derniers sont utilisés. Des modèles d'apprentissage tels que l'apprentissage par problème, la pédagogie de projet, l'apprentissage par découverte etc. combinés à des dispositifs d'apprentissage innovant peuvent significativement accroître la complexité des processus d'enseignement et d'apprentissage (Jochems, van Merriënboer et al. 2003). Une certaine volonté de favoriser des savoirs plus profonds et opérationnels au travers d'activités « riches » amènent à scénariser de manière précise les situations d'apprentissage instrumentées. Comme le rappelle Schneider & al (2003) :

« Il ne suffit pas de placer les sujets dans des situations de construction et de collaboration pour qu'un apprentissage efficace ait lieu. En règle générale l'enseignant doit mettre en place des scénarios structurés pour obtenir ce résultat. Il doit chercher un équilibre harmonieux entre la liberté nécessaire au développement intellectuel et certains principes de guidage. D'un côté, il faut favoriser un véritable engagement dans un projet et les confrontations d'idées (donc traiter un apprenant comme un "petit chercheur") et de l'autre il faut structurer et soutenir les activités et exercer un certain contrôle. » (p.4)

¹ Business Process Management

² Services Oriented Architecture

INTRODUCTION

Des deux constats précédent découlent un grand nombre de tâches à effectuer que ce soit du côté de l'équipe pédagogique que du côté apprenant. Beaucoup de ces tâches sont effectuées à travers la médiation des TICE, certaines de manière automatisée par le système, d'autres non. Le nombre des tâches et leurs interrelations augmentent sensiblement la complexité de gestion d'un dispositif. Pour autant cette complexité se doit de pouvoir être contrôlée et gérée dans une perspective respectant les intentions pédagogiques de l'enseignant. Classiquement, cette complexité est rarement abordée de front. Il n'est pas rare d'observer des dispositifs focalisés sur un usage important de plateformes de formation (Learning management system – LMS) même lorsque ces dernières ne répondent que partiellement aux intentions pédagogiques. Sur ce point notre pensée rejoint celle de Georges-Louis Baron (2012) écrivant :

« Les plates-formes mises à disposition des enseignants sont puissantes, mais ce sont aussi des monuments de complexité, dont le comportement dépend d'un grand nombre de variables d'environnement. Elles sont d'un usage commode lorsqu'il s'agit de déposer des ressources, voire de gérer le dépôt de projets par les apprenants. Leur appropriation est en revanche bien plus délicate quand il s'agit de planifier des activités (et a fortiori de concevoir des scénarios précis), ce qui exige classiquement, en formation à distance, l'intervention d'une équipe. » (p.117).

Évoquant la nécessité de recentrer la conception de dispositif d'apprentissage autour de scénario, l'auteur ajoute :

« L'enjeu me semble désormais être le développement et la diffusion de nouveaux types d'outils, permettant aux enseignants de concevoir et de mettre en œuvre à leur gré des ressources et des séquences d'activités ne visant pas forcément à être ensuite mutualisées. Cela permettrait de faciliter, sous réserve de l'organisation de formations spécifiques et de la constitution de communautés de pratique, l'émergence de nouvelles formes d'instrumentation des processus d'enseignement, tirant mieux parti des possibilités contemporaines de l'Internet pour favoriser les apprentissages des jeunes générations. » (p.118)

Dans ce mémoire notre proposition vise à proposer une piste pour le développement de dispositifs d'apprentissage basés sur les scénarios. Les langages de modélisation pédagogique tels qu'IMS-LD développés en réponse à cette problématique ont ouvert la voie mais ne rencontrent pourtant pas le succès auquel ils aspirent. Les spécifications sont souvent complexes et les ingénieurs pédagogiques passent beaucoup de temps et d'effort juste pour être en mesure de définir un scénario pédagogique (De Vries & al, 2005 ; Morales & al. 2008). De plus, la mise en œuvre de la spécification dans les environnements d'apprentissage en ligne est encore entravée de nombreuses difficultés et les outils faciles d'utilisation manquent cruellement. Enfin nous faisons le constat que ces langages sont issus du domaine de l'éducation et restent globalement inconnus des autres acteurs avec qui les enseignants et plus particulièrement les ingénieurs pédagogiques sont amenés à collaborer.

Notre approche prend le parti inverse. Nous investiguons la possibilité d'utiliser dans le domaine de l'éducation une technologie déjà utilisée par l'ingénierie logicielle et connue du « monde de l'entreprise ». Notre problématique vise à offrir un langage commun aux différents acteurs impliqués dans la mise en œuvre de dispositifs de formation, afin de respecter les intentions pédagogiques de départ tout au long du processus de conception. Notre hypothèse est que le langage BPMN 2.0 pourrait être dans cette optique une solution intéressante. En effet ce langage est une notation abstraite ayant pour objectif d'être facilement compréhensible et d'offrir un pont standardisé entre tous les acteurs (techniques ou non) impliqués dans le développement d'applications de services. De plus, BPMN 2.0 tend à s'imposer comme un langage exécutable incontournable en ingénierie des processus métiers. L'enjeu est donc de savoir en quoi la logique de « workflow » propre à ce langage est compatible avec celle d'un scénario pédagogique et ce que pourrait apporter l'utilisation d'un tel standard à l'ingénierie pédagogique. Notre exploration nous a amené à utiliser des outils de développement spécialisés dans la conception d'infrastructures BPM (les BPMS) afin de tester l'implémentation de scénarios pédagogiques. Nous avons par ce biais déployé et modifié une plateforme standard de gestion de workflows que nous avons intitulé « E-flow » et sur laquelle nous avons effectué nos tests.

2. STRUCTURE DU MÉMOIRE

Le *premier chapitre* (facultatif pour les lecteurs avertis) est un état de l'art des grands modèles théoriques de l'apprentissage. Ces derniers sous-tendent les pratiques pédagogiques et un retour sur leur histoire nous rappelle combien « apprendre » est un processus complexe ne pouvant être laissé au hasard. Ce panorama permet selon nous de ne pas perdre de vue les présupposés fondamentaux de chacune des approches et évite de nous enfermer dans une vision trop « techno-centrée » de l'éducation avec les TICE. De plus, il nous permet d'expliciter en conclusion notre vision des TICE en tant que soutien aux processus d'apprentissage.

Au *second chapitre* nous introduirons la scénarisation pédagogique. Nous définirons brièvement, l'Instructional Design dont nous considérons les modèles comme soutien à la conception de scénario. Nous présenterons ensuite une définition selon Pernin et Lejeune (2004) sur laquelle nous appuierons notre vision de la scénarisation. Enfin, nous présenterons la Théorie de l'Activité que nous prenons comme un cadre conceptuel important pour la modélisation pédagogique avec BPMN. Ces trois présentations nous permettront d'introduire notre conception de la scénarisation sous l'angle d'un « workflow » orchestrant le processus d'apprentissage dans un environnement techno-pédagogique.

Au *troisième chapitre* nous aborderons l'ingénierie pédagogique au regard de la modélisation. Nous présenterons l'évolution de la notion d' « objet d'apprentissage » à travers la définition des normes et standards d'ingénierie pédagogique ainsi que la vision CSCL de conception de scripts formalisant les activités collaboratives. Nous dresserons ensuite un panorama des différents travaux sur les langages de modélisation pédagogique afin de mieux situer BPMN. Nous introduirons également la notion « d'ingénieur pédagogique » définissant aujourd'hui un acteur important des dispositifs de formation instrumentés vers lequel nous destinons les propos du mémoire. Enfin nous introduirons une réflexion sur les enjeux d'une ingénierie des EIAH³ dirigée par les modèles et basée sur une architecture SOA⁴ pour la génération d'environnements adaptés aux scénarios pédagogiques (et non l'inverse).

Au *chapitre 4* nous présenterons brièvement le langage BPMN 2.0 et l'approche BPM⁵ dans laquelle il s'inscrit. BPMN est un langage graphique récent pour la description de workflow (Business Process Modelling and Notation). Nous introduirons son concept, les bases nécessaires à la compréhension de ce langage ainsi que les outils existant pour son utilisation. Une extension de ce chapitre est disponible en annexe apportant un complément d'information technique sur le fonctionnement de la spécification BPMN 2.0.

Le *chapitre 5* abordera concrètement l'utilisation de BPMN sous un paradigme éducatif. Nous présenterons au cours de ce chapitre une conception de dispositifs basés sur (1) des scénarios pédagogiques formalisés en BPMN, (2) une plateforme d'exécution de workflow et (3) une architecture orientée services. Nous y définissons une palette d'éléments graphiques apte selon nous à représenter des « workflows pédagogiques » en accord avec le cadre théorique développé au long de ce mémoire, et testons le déploiement une plateforme simple pour l'exécution de scénarios pédagogiques.

Chaque chapitre dispose d'une page de conclusion résumant les informations principales au regard de notre problématique.

³ Environnement Informatique pour l'Apprentissage Humain

⁴ Services Oriented Architecture

⁵ Business Process Management

CHAPITRE I. DES THÉORIES ÉDUCATIVES À LA SCÉNARISATION PÉDAGOGIQUE

« In E-Learning, the emphasis is often on technical aspects, and the reason for using technology – the desired pedagogical added-value – is in danger of being neglected.[...] Many educators believe that information and communication technologies fundamentally transform and improve the process of learning. This assumption is not new: Each major step in the evolution of modern technology gave raise to new hopes and promises. However, the new technologies have, in most cases, not had the proclaimed effect. »

[Reichert & Hartmann, 2004, p. 1]

Les sciences de l'éducation puisent leurs fondements dans divers champs théoriques tels que l'épistémologie, la psychologie, la sociologie, la philosophie et les sciences cognitives. Cette diversité est à la base de différentes théories de l'enseignement et de l'apprentissage qui tentent de modéliser ce qu'est enseigner et ce qu'est apprendre, proposant chacune un certain rôle de l'enseignant, une certaine vision des activités de scénarisation. Dans le contexte des technologies numériques se pose dès lors la question de la scénarisation pédagogique et la mise en œuvre d'un modèle d'apprentissage dans la conception des environnements informatiques pour l'apprentissage humain (EIAH).

Ce chapitre propose un état de l'art sur les quatre modèles principaux qui sous-tendent les pratiques éducatives. Nous poserons ainsi progressivement les bases du second chapitre où nous nous intéresserons plus spécifiquement aux questions de la mise en pratique de ces modèles à travers l'Instructional Design et la scénarisation pédagogique.

INTRODUCTION

Dans le domaine de l'ingénierie des dispositifs, la notion d'apprentissage est parfois reléguée à l'arrière-plan, se masquant comme un allant-de-soi derrière une vision très « techno-centrée ». Un retour dans l'histoire des théories de l'apprentissage nous rappelle combien « apprendre » est un processus complexe qui ne peut, pour un formateur, être laissé au hasard. La conception d'un dispositif pédagogique se doit d'être ancré dans une vision pédagogique plus large afin, bien entendu, de justifier les choix techniques, mais aussi, peut-être, de ne pas perdre de vue les présupposés fondamentaux de la démarche qu'il a choisi d'adopter. L'histoire de la pédagogie n'est bien entendu pas neuve, les énigmes entourant les capacités d'apprentissage de l'être humain animent les débats depuis l'antiquité. Il existe différents courants de pensées dont nous accordons les origines à deux visions extrêmes de l'acte d'apprendre : le rationalisme et l'empirisme (Crahay, 1999, p. 13). S'il existe de très de nombreuses classifications⁶, l'objectif ici n'est pas de faire un inventaire exhaustif des différents courants et théories de l'apprentissage, mais de présenter les modèles les plus largement utilisés. Au-delà du modèle transmissif⁷, une majorité de théoriciens en éducation s'accordent pour identifier les modèles de l'enseignement et de l'apprentissage selon quatre courants majeurs:

⁶ Bertrand (1979), Bertrand & Valois (1980) répertoriaient déjà il y a 30 ans plus de 80 classifications des modèles d'apprentissages. Voir annexe 1 pour une proposition de classification chronologique.

⁷ Conception de l'apprentissage classique héritée des pédagogies traditionnelles, où l'enseignant détient, organise, structure le savoir

- *Le courant behavioriste* conceptualisé par des auteurs tels que Pavlov (1901, 1927), Watson (1913) Thorndike (1922, 1932) ou Skinner (1957).
- *Le courant cognitiviste* représenté entre autres en science de l'éducation par les psychologues nord-américains Gagné (1985, 1988, 1992), Ausubel (1960, 1968) ou encore Tardif (1992).
- *Le courant constructiviste* développé par Piaget (1923, 1936, 1970) dans son approche développementale, Bruner (1966) sous un angle interactionniste et Vygotski (1962, 1978) dans son approche historico-culturelle.
- *Le courant socioconstructiviste* s'appuyant sur les travaux des trois auteurs précédent, ce courant est représenté par des auteurs européens tels Perret-Clermont(1979), Doise et Mugny (1981, 1997), ou encore Gilly (1989, 1995) ou américains tels que Bruner (1991), Brown, Campione, Gardner et Perkins (1995).

Nous présentons brièvement ci-dessous, chacun des courants en tentant de les lier à leurs applications actuelles dans le domaine de l'ingénierie des EIAH.

1. MODÈLE BÉHAVIORISTE

Le bémorisme est une approche en psychologie qui consiste à s'appuyer sur le seul comportement observable de façon à caractériser comment il est déterminé par l'environnement et les interactions de l'individu avec son milieu, sans faire appel à des mécanismes internes au cerveau ou à des processus mentaux non directement observables⁸.

1.1 Origine d'une approche comportementale.

Le terme « bémorisme » fut initié par Watson (1913) s'inspirant des travaux du prix Nobel Petrovitch Pavlov (1901) sur le conditionnement classique (ou conditionnement répondant). Ce dernier démontra grâce à des observations sur les glandes salivaires du chien qu'il était possible d'associer par répétition et proximité temporelle, un stimulus inconditionnel (production de salive à la vue de nourriture) à un stimulus conditionnel (faire sonner une cloche lors de l'apport de nourriture) et ainsi d'obtenir une réponse conditionnelle (faire saliver le chien lorsque l'on fait sonner la cloche). On découvrit ainsi certains mécanismes fondamentaux de l'apprentissage à l'œuvre chez l'humain dans la formation des émotions et attitudes ou dans de nombreux phénomènes sociaux (Crahay, 1999, p. 122).

1.2 Adaptation à l'apprentissage humain.

Généralisant à l'apprentissage humain les résultats des recherches sur la psychologie animale, des auteurs tels que Thorndike ou Skinner – parfois qualifiés de néo-behavioriste – vont développer des théories de l'apprentissage qu'ils appliqueront dans des programmes d'enseignement. Ainsi Thorndike (1922) conceptualisera les premières leçons *drill and practice* basées sur une notion d'apprentissage graduel par essai et erreur. Le savoir est découpé en différents composants qu'un élève précédemment introduit à la tâche (loi de préparation) assimile à travers la répétition (loi de l'exercice) rapidement suivie d'un feedback positif ou négatif (loi de l'effet). Dans cette vision de l'apprentissage, le rôle de l'enseignement est de récompenser les comportements jugés appropriés face à une tâche et d'éliminer les comportements jugés incorrects, moyennant suffisamment de répétitions et de rétroactions pour chaque élève. Nous retrouvons ici la notion de progression individualisée, chère aux psychologues behavioristes, en rupture avec les approches transmissives, où des cohortes d'apprenant progressent exactement au même rythme.

et le transmet à l'apprenant. On retrouve les fondements de ce mouvement dans l'empirisme anglais avec notamment les travaux de John Locke (XVII^{ème}).

⁸ Le behaviorisme se forme notamment en réaction aux théories mentalistes de l'époque qui s'oriente sur l'étude des phénomènes psychiques plutôt que sur l'étude des comportements.

Avec les travaux de Skinner sur le conditionnement opérant et l'enseignement programmé, le courant behavioriste en sciences de l'éducation inspirera le développement des machines à enseigner. Pour Skinner la matière à enseigner doit être fragmenté avec le niveau de granularité le plus fin possible afin que chaque point soit facilement compréhensible avant d'aller plus loin. Les tâches sont ainsi réparties en sous-tâches considérées comme un objectif d'apprentissage distinct axé sur des comportements observables (behaviours). Contrairement à la vision de Thorndike, pour qui les récompenses et punitions contribuent également à renforcer le comportement souhaité. Pour Skinner seules les renforcements positifs sont dignes d'intérêt en éducation. « *Nous ne renforçons pas la bonne prononciation en punissant la mauvaise, ni des mouvements habiles en punissant la maladresse* » (Skinner, 1968, p. 178). Ainsi selon lui, les séquences d'apprentissage doivent être programmées de manière à ce que l'élève puisse s'approprier l'essentiel des compétences de manière graduelle et à son rythme, en réalisant le minimum d'erreurs et en obtenant le maximum de retour sur l'exactitude de sa réponse. Pour Skinner, « *enseigner n'est rien d'autre, en effet, qu'arranger les conditions de renforcement dans lesquelles les élèves apprendront* » (ibid., p. 77).

1.3 Un héritage important malgré de nombreuses critiques.

Cette vision déterministe d'un enseignement segmenté et planifié avec précision, d'une progression différenciée selon les élèves où les rétroactions et les régulations sont primordiales pour l'apprenant, inspireront la pédagogie de la maîtrise (Bloom, 1968), différentes théories cognitivistes telles que la théorie des conditions d'apprentissage (Gagné, 1974), les premières machines à enseigner et les systèmes d'enseignement assistés par ordinateur (EAO) en général. Les exerciceurs, leçons informatisées, quizz, tests font parties des fondamentaux du modèle e-learning encore à l'œuvre aujourd'hui. Pourtant, comme le rappelle Richelle (1977), le behaviorisme et en particulier l'œuvre de Skinner, ont soulevé d'importantes critiques. Ces critiques (niant parfois aveuglément les apports importants du behaviorisme à la psychologie contemporaine) sont d'ordre:

- *Philosophique et idéologique* : deshumanisation de l'enseignement, vision mécaniste de l'apprentissage, fonctionnement de la pensée déterministe remettant en cause les notions de mérite et de libre-arbitre (Crahay, 1999, p. 169) ;
- *Cognitive et pédagogique* : sous-estimation des capacités sémiotiques du cerveau humain (Chomsky, 1959), fragmentation du savoir empêchant la visualisation du concept dans son ensemble (théorie gestaltiste), faible prise en compte des enjeux motivationnelles dans l'apprentissage, non-prise en compte des processus cognitifs (le cerveau est une « boîte noire ») .

En réaction, de nouvelles théories voient le jour qui, elles, donnent au contraire une place majeure aux aspects non observables de l'activité mentale : le constructivisme, en Europe, à partir des années 30 et le cognitivisme, aux États-Unis, dans les années 60. Ces modèles seront présentés dans les points suivants.

2. MODÈLE CONSTRUCTIVISTE

Si le modèle behavioriste axait son raisonnement sur l'environnement d'apprentissage et les comportements observables, le constructivisme va au contraire repositionner l'apprenant au centre, en postulant que l'apprenant construit activement ses connaissances au travers de ses interactions avec le réel et l'assimilation avec ses connaissances antérieurs. Pour Perrenoud (2003) :

Le constructivisme n'est ni une mode, ni une doctrine. Ce n'est pas non plus, en soi, une démarche pédagogique. C'est une " loi " de l'apprentissage humain, qui dit que tout apprentissage passe par une activité mentale du sujet, une activité de réorganisation du système de schèmes et de connaissances existant. Sans cette activité, invisible mais intense, aucun élément nouveau ne peut être intégré. (p. 7)

2.1 Approche selon l'épistémologie génétique.

Le constructivisme est basé sur les travaux de Jean Piaget, épistémologue, dont l'œuvre immense a eu pour but « *d'élaborer par les méthodes scientifiques, une explication biologique de la connaissance humaine* » (Crahay, 1999, p.173). Au-delà de ses travaux sur les stades développementaux des enfants, Piaget décrit une théorie du fonctionnement cognitif. Pour lui, la connaissance ne s'impose pas à un sujet passif (modèle transmissif) ; c'est l'apprenant qui va à sa rencontre et s'efforce de l'intégrer (ce que Piaget nomme le processus d'*assimilation*), ou en cas de conflit cognitif, tente de l'ajuster à ses schèmes d'action ou de pensée (processus d'*accommodation*). Voir figure 1 ci-dessous.

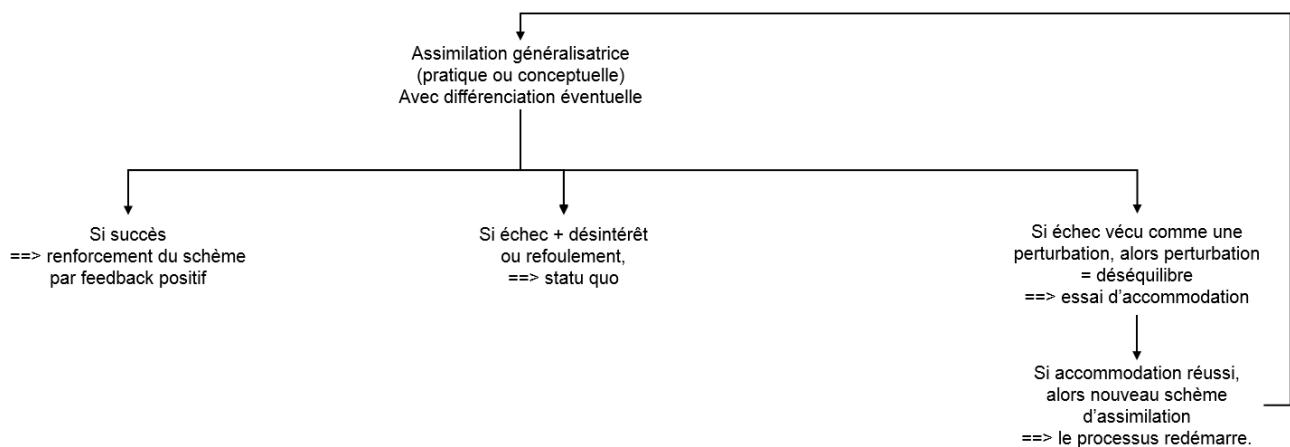


Figure 1: Processus d'assimilation et accommodation selon Piaget (Crahay, 1999, p. 179)

Ainsi selon le modèle constructiviste, l'acquisition de connaissance ne se réalise pas par simple empilement mais passe par une réorganisation de conceptions mentales précédentes, un travail de construction ou de reconstruction. Pour Piaget, l'assimilation et l'accommodation forment un couple indispensable à l'activité cognitive dont les différents processus d'équilibration seront développés dans *l'équilibration des structures cognitives* (1975).

2.2 Approche historico-culturelle.

Une autre figure majeure du constructivisme se distingue en la personne de Lev Vygotski, psychologue contemporain de Piaget, bien que ces travaux ait été découverts bien plus tard. Vygotski apporte une approche historico-culturelle à la pensée constructiviste. L'expérience du monde par l'apprenant est indissociable des interactions sociales et appropriations culturelles dans lesquelles il se trouve. Pour Vygotski, l'appropriation progressive des construits culturels se subordonnent aux développements cognitifs de l'apprenant. On ne peut comparer la conception du monde d'un enfant de l'âge de pierre, du Moyen Âge et du XX^{ème} siècle car celle-ci s'opère via la médiation de construits culturels (Langage, stylo, ordinateur...) et l'interaction sociale propre à une

époque (Schneuwly & Bronckart, 1985, p. 35). La notion de *zone proximale de développement*⁹ qu'il développe représente une synthèse conceptuelle de l'apport des conceptions vygotskiennes à la connaissance de la dynamique du développement cognitif. Les travaux de Vygotski sont à l'origine des théories socioconstructivistes que nous aborderons plus particulièrement au point 1.4.

2.3 Apports du constructivisme à la pédagogie moderne.

Le constructivisme a profondément marqué les pédagogies modernes, comme le souligne Perrenoud (2003) « *nombre de systèmes éducatifs ont fait du constructivisme leur "théorie de référence"* » (p. 9). Le constructivisme fait aujourd'hui partie de l'outillage théorique de base des acteurs de l'enseignement notamment à travers les travaux de J-S Bruner, qui s'appuyant sur les travaux de Piaget et Vygotski, assoira les fondements modernes du constructivisme. Doolittle (1999) résumera 8 conditions nécessaires à une pédagogie constructiviste, présenté ci-dessous.

1. *Présenter des situations d'apprentissage complexes renfermant des activités authentiques.* Les élèves doivent résoudre des problèmes complexes similaires à ceux qu'ils rencontrent dans la réalité
2. *Procurer des interactions sociales.* La collaboration entre les apprenants procure des occasions de partager leur manière de concevoir les choses
3. *Un contenu et un savoir-faire signifiants à l'élève.* Le contenu et les habiletés qu'il se propose d'acquérir doivent avoir un sens pour lui
4. *Un contenu et un savoir-faire proches des acquis de l'élève.* Tout apprentissage doit partir des acquis de l'élève
5. *Les élèves doivent bénéficier d'une évaluation formative continue.* Comme l'apprentissage est un processus continu, l'enseignant doit régulièrement évaluer de façon formative l'état de l'apprentissage actuel de l'élève en tenant compte du processus plus large dans lequel il s'insère.
6. *Être responsable de son apprentissage.* L'apprentissage de l'apprenant dépend de lui-même et non de son enseignant. Cet apprentissage nécessite un engagement profond de la part de l'apprenant.
7. *Les enseignants sont d'abord des guides et des agents facilitants de l'apprentissage.* L'enseignant n'est pas un instructeur. Il propose plutôt à l'élève des expériences aptes à susciter des acquisitions significatives.
8. *Revoir des contenus et les présenter selon diverses perspectives.* La connaissance n'est pas unique. L'élève apprend à percevoir une réalité selon divers points de vue afin de construire sa propre vision de cette réalité perçue.

2.4 TICE et constructivisme.

Dans ce domaine, force est de constater que le mariage entre constructivisme et technologies numériques n'est pas évident. Si des auteurs tels que Matusevich (1995) affirme en évoquant le constructivisme et les technologies numériques « *This link is so strong that it will cause a fundamental shift away from didactic techniques to a unifying constructivist framework* » (p.1 - abstract) ; rappelons que la technologie n'est pas constructiviste "en soi", et s'est avant tout la façon de l'utiliser qui importe. En positionnant l'apprenant en tant qu'acteur principale d'un savoir construit en interaction avec son environnement physique et social, le constructivisme va plus que jamais mettre en avant la notion d'activité scénarisée. Le constructivisme fournit alors un cadre de référence au développement de pédagogies dans lesquelles les technologies peuvent apporter une contribution intéressante. Comme le rappelle Jonassen les TICE offrent les potentialités nécessaires pour que « *Les apprenants fonctionnent comme des concepteurs utilisant la technologie comme outils pour analyser le monde, accéder à l'information, interpréter et organiser leurs connaissances personnelles, et représenter ce qu'ils savent à d'autres* » (p. 2).

⁹ Postule que le potentiel d'apprentissage d'un individu est fonction non pas de ce qu'il est capable de réaliser seul, mais en collaboration avec un adulte ou un individu plus expérimenté.

L'influence du constructivisme est particulièrement prégnante dans le développement des problèmes assistés par ordinateur (« Problem based learning ») ou des **micro-mondes**¹⁰. Citons des micro-mondes à caractère général comme Logo (Papert, fin des années 70) et plus récemment Squeak (Guzdial & Rose, 2001) ou spécialisé dans une discipline comme ScienceSpace (Dede, Salzman & Loftin, 1996) pour la physique et Cabri-géomètre pour la géométrie (Laborde & Capponi, 1994). Des outils qui doivent aussi beaucoup à un autre grand courant des théories de l'apprentissage, le cognitivisme, que nous présentons ci-après.

3. MODÈLE COGNITIVISTE

3.1 Aux origines du traitement cérébral de l'information.

Le cognitivisme en sciences de l'éducation trouve son origine dans les années 1950-1960, en réaction au behaviorisme (voir notamment les travaux en psychologie linguistique de Chomsky) et en parallèle avec le développement de l'informatique et de l'intelligence artificielle. Le cognitivisme, comme son nom l'indique va s'intéresser à l'étude de la cognition, soit le fonctionnement de l'intelligence, l'origine de nos connaissances ainsi que les stratégies employées pour assimiler, retenir et réinvestir ces connaissances. Le cognitivisme s'intéresse essentiellement à la perception, le traitement en mémoire, le langage et ce, en regard du fonctionnement du cerveau (Legendre, 1993 ; Tardif 1992).

Pour les cognitivistes, mettant en parallèle l'intelligence et le fonctionnement d'un ordinateur, les hommes peuvent être considérés « *comme des ensembles systémiques ouverts qui ont la possibilité de communiquer avec l'environnement [...] traitent l'information venant de l'extérieur et se régule en fonction d'elle* » (Crahay, 1999, p. 251). De manière plus fondamentale encore, la pensée humaine peut être comparée à un système computationnel qui manipule des symboles, les analyse et les emmagasine en mémoire pour les réutiliser dynamiquement. Les théories cognitivistes s'intéressant au traitement de l'information partent du principe que le cerveau est basé sur trois éléments fondateurs : La perception sensorielle (processus de capture de l'information par nos 5 sens, avec éventuellement un court traitement), la mémoire de travail (stockage à court terme de plus ou moins 7 informations ou groupe d'informations pour l'analyse) et la mémoire à long terme (stockage de l'information pouvant ainsi être réutilisé ultérieurement).

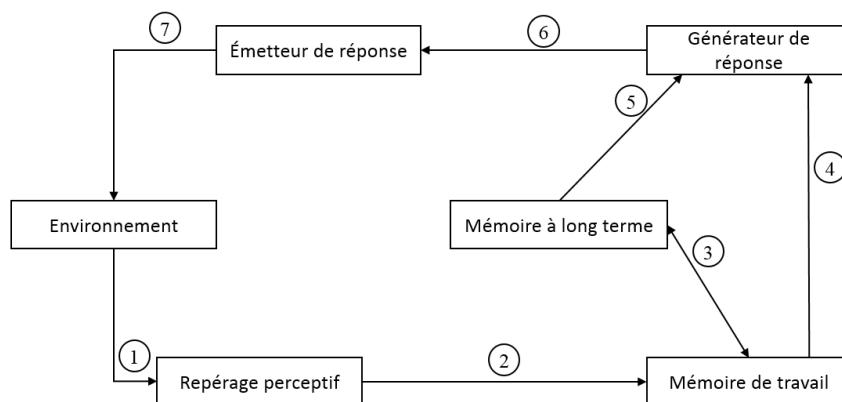


Figure 2: Principe simple du traitement de l'information selon Ellen D. Gagné (1985)

¹⁰ « L'idée essentielle sous-jacente à la notion de micromonde est de créer un environnement, respectant différentes contraintes de construction, avec lequel d'autres personnes pourront exercer leur créativité. » (Bruillard, 1997, p. 140) ; « Modèle réduit d'un monde réel ou abstrait utilisé comme environnement d'apprentissage. » (Legendre, 1993, p. 849).

Ainsi apprendre revient, pour les cognitiviste, à intégrer l'information nouvelles en mémoire à long terme (Lindsay et Norman, 1980, p. 492).

3.2 Apports à l'apprentissage.

Des auteurs tels que Robert M. Gagné, David Ausubel ou Jacques Tardif vont s'intéresser au traitement de l'information par le cerveau et en déduire des implications directes pour l'apprentissage. Gagné (1985) propose ainsi une théorie des conditions d'apprentissage dans laquelle il postule qu'il existe plusieurs types et niveaux d'apprentissage. Il distingue ainsi 5 catégories principales : l'information verbale, les capacités intellectuelles, les stratégies cognitives, les capacités motrices, les attitudes. Chacune de ses catégories nécessite différents types d'instructions, Gagné définit 9 instructions (externes) et les processus d'apprentissage liés (internes) :

1. *gagner l'attention:* (réception)
2. *informer l'apprenant de l'objectif:* (espérance « expectancy »)
3. *stimuler le rappel d'une connaissance acquise précédemment:* (récupération « retrieval »)
4. *présenter le matériel-stimulus:* (perception sélective)
5. *guider l'apprenant:* (encodage sémantique)
6. *faire ressortir les performances:* (réagir)
7. *produire un feedback sur les performances:* (renforcement)
8. *évaluer les performances:* (récupération « retrieval »)
9. *augmenter la rétention et le transfert:* (généralisation)

Selon l'auteur, ces instructions devraient satisfaire ou fournir les conditions nécessaires à l'apprentissage et servir de base pour la conception de l'enseignement et le choix des supports appropriés (Gagné, Briggs et Wager, 1992). Tardif (1992) propose pour sa part un modèle d'apprentissage basé sur les notions de stratégies cognitives. Ce modèle est fondé sur l'importance de l'appropriation progressive des stratégies cognitives et métacognitives (générales et spécifiques) jugées nécessaires à la démarche d'apprentissage. Tardif cherche à favoriser l'engagement cognitif et affectif (motivation scolaire) en montrant à l'apprenant comment traiter les informations et en l'amenant à effectuer des transferts. Il définit ainsi 5 principes de base de l'apprentissage (*ibid.*) :

1. L'apprentissage est un processus dynamique de construction des savoirs : sujet actif, constructif et motivé.
2. L'apprentissage suppose l'établissement de liens entre les nouvelles informations et celles déjà organisées (représentations).
3. L'apprentissage exige l'organisation incessante des connaissances.
4. L'apprentissage suppose la mobilisation de stratégies cognitives et métacognitives ainsi que des savoirs disciplinaires.
5. L'apprentissage produit renvoie aux connaissances déclaratives, procédurales et conditionnelles.

3.3 TICE et cognitivisme

L'approche cognitive a eu un impact non négligeable sur le développement des TICE, naviguant parfois entre espoir et désillusion. Le cognitivisme donnera lieu à d'importants travaux dans le domaine éducatif sur la façon la plus adéquate de présenter le savoir notamment grâce aux possibilités hypertextes et hypermédias de l'informatique. Associé à une vision constructiviste, le cognitivisme sera à la base du développement des cartes conceptuelles interactives, des simulations et de l'enseignement intelligemment assisté par ordinateur (EIAO). De plus le cognitivisme aura un impact important dans le domaine de l'ergonomie des EIAH, notamment à travers, les travaux sur les phénomènes de surcharge cognitive et les modalités de la perception sensorielle. Cependant les théories cognitivistes, en se focalisant principalement sur les processus internes à l'apprenant, ont peu pris en compte la dimension sociale dans le développement du raisonnement de l'apprenant.

4. MODÈLE SOCIOCONSTRUCTIVISTE

Si pour certains auteurs le socioconstructivisme n'est qu'un des visages du constructivisme, nous avons choisi dans ce mémoire de lui consacrer une section particulière, d'une part car cela respecte une certaine culture européenne (pour ne pas dire genevoise) des sciences de l'éducation ; d'autre part car la multiplication des travaux de ces vingt dernières années prenant ouvertement pour cadre le socioconstructivisme tendent à l'ériger comme un modèle à part entière. Ainsi si le socioconstructivisme plonge ses racines dans le constructivisme, il s'en démarque par la prise en compte des aspects culturels et des interactions sociales multiples dans la construction des savoirs. Nous pouvons grossièrement distinguer deux approches du socioconstructivisme dans la littérature. Une première approche très interactionniste, caractéristique d'auteurs européens tels que Perret-Clermont, Gilly, Doise et Mugny s'attachant à étudier les interactions sociocognitives à la lumière du constructivisme Piagétien. Une seconde approche qualifiée de psychologie culturelle, dans la droite lignée des travaux de Vygotski et Bruner, caractérisée par des auteurs américains tels que Brown, Campione, Perkins etc.

4.1 Approche européenne – Évolution de l'interactionnisme Piagétien.

Pour les tenants de la première approche, il est fondamentale de sortir d'une vision duale de l'apprentissage « individu-tâche » et de passer à une vision ternaire de l'apprentissage « Individu-collectif-tâche ». Pour ces auteurs l'action du sujet sur le réel est rarement une activité strictement personnelle en totale autonomie. Dès lors il convient de revoir l'acquisition des schèmes de pensée et les processus d'équilibration et d'accommodation des structures mentales sous le paradigme d'une interaction sociale avec le réel. Anne-Nelly Perret-Clermont (1979) démontre que l'interaction sociale ne se résume pas à des processus d'imitation et d'étayage, mais bien à une véritable construction du savoir par interaction. À travers des expériences classiques de Piaget (conservation des liquides, des nombres et des longueurs) revisitées sous une perspective collective elle révèle des progrès significativement supérieurs d'enfants ayant agi avec un partenaire par rapport à des enfants ayant agi seuls, et ce, quel que soit le niveau de compétence du partenaire. Ces travaux vont mettre en exergue le rôle du « conflit sociocognitif » dans la construction des savoirs. D'autres études regroupées sous ce titre donneront une sérieuse assise expérimentale à ce concept (Doise 1993 ; Doise et Mugny, 1981; Doise et Mugny, 1997). Ces auteurs vont ainsi s'attacher à étudier les conditions nécessaires pour qu'un conflit sociocognitif permette effectivement d'obtenir un développement cognitif (apprentissage) du sujet au sein d'un groupe. L'apprentissage est pour eux un mécanisme de causalité circulaire qui progresse en spirale (Doise et Mugny, 1981, p.173). Le développement sociale va engendrer un développement cognitif permettant au sujet de participer à des interactions sociales plus élaborés et ainsi de suite.

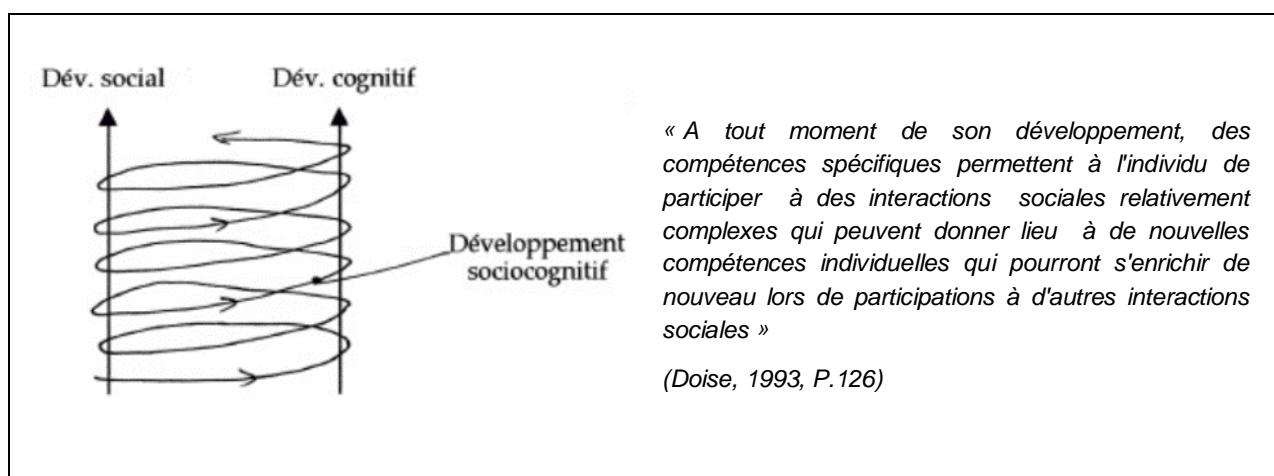


Figure 3: Développement sociocognitif en spirale (Doise, 1993)

Doise résume ces observations à 6 propositions fondamentales (1993, P. 127-128) :

1. C'est en coordonnant ses démarches avec celles de partenaires que le sujet est conduit à des équilibrations cognitives dont il n'est pas capable individuellement.
2. Les enfants peuvent tirer un profit personnel après avoir participé à ce type d'échanges, ils sont capables d'effectuer seuls les tâches réussies préalablement en situation sociale.
3. Les échanges interindividuels deviennent source de progrès cognitif par les conflits sociocognitifs qu'ils suscitent. Il n'est pas indispensable qu'un des partenaires de l'interaction soit porteur du modèle correct de résolution du problème.
4. Les bénéfices des interactions dépendent (comme dans tout apprentissage) des compétences initiales des sujets.
5. Certaines situations-problèmes marquées socialement peuvent générer un conflit sociocognitif, producteur de réorganisation des connaissances.
6. Le processus interactif est moteur du développement cognitif, et il y a une causalité circulaire entre compétences cognitives et communicationnelles.

Avec l'approche européenne du socioconstructivisme, jamais les mécanismes d'interactions « groupe-individu » et leurs impacts sur l'apprentissage personnel n'auront été étudiés aussi finement. Rappelons toutefois que si ces travaux soulignent l'importance de la conversation et de la confrontation dans les dynamiques de construction de connaissances individuelles, ils en dessinent également les limites. En effet, il ne suffit pas de mettre les individus en interaction pour qu'un apprentissage ait lieu. Un écueil que l'on retrouve aussi bien dans l'enseignement que dans le développement d'outil e-learning se réclamant du socioconstructivisme. Comme nous avons pu le voir, le socioconstructivisme postule avant tout que l'apprentissage survient lorsque des individus tentent de coordonner des centrations cognitives divergentes dans la perspective d'une action commune concrète. Les conditions nécessaires à ce processus ne vont bien entendu pas de soi. Le rôle de l'enseignant est dès lors primordial, non plus comme celui qui transmet le savoir, mais comme un « chef d'orchestre » scénarisant avec précision les activités individuelles et collectives, en vue de provoquer les apprentissages.

4.2 Approche américaine – Poursuite des travaux de l'école russe.

Bien que l'environnement social de l'apprenant soit composé d'individus en interaction, il est aussi constitué d'un milieu physique, d'un contexte culturel. L'approche socioconstructiviste précédente se concentre sur les processus cognitifs indépendants de l'environnement d'apprentissage. C'est à présent, sur ce dernier, que va se centrer l'approche culturelle du socioconstructivisme. Résolument plus proche de Vygotski que de Piaget, ces défenseurs s'intéressent à la construction sociale des savoirs avec une centration particulière sur le milieu (les structures sociales, la culture, les individus et les outils). Poursuivant la réflexion de Vygotski¹¹, les tenants de cette approche considèrent le savoir humain comme une immense construction collective se transmettant par la culture et les outils techniques propres à une époque. Ainsi pour Perkins (1995), la connaissance d'un individu ne se résume pas à celle se trouvant dans sa mémoire, elle se définitie plutôt comme à un ensemble constitué du cerveau et de tous les outils (livres, ordinateurs, plans etc.) qu'il est capable de mobiliser face à une situation donnée. De plus selon lui, les connaissances « peuvent aussi être répartis entre plusieurs individus dont chacun ne dispose de toutes les ressources » (p.57). Perkins définit de cette manière le concept d'*individu-plus* qu'il oppose à *l'individu-solo*. L'intelligence est ainsi distribuée au sein d'une société au-delà de l'organisme lui-même. Partant de ce principe en rupture totale avec une vision transmissive de l'enseignement, les apprenants tout comme l'enseignant sont porteurs d'une part de savoir. Des auteurs tels que Brown et Campione (1995) vont dès lors mener des expérimentations sur l'enseignement réciproque au sein de communautés d'apprenant. Pour eux, « la classe peut être envisagé comme un lieu de partage des savoirs, tout autant qu'un centre de ressources au sein duquel sont

¹¹ Voir présentation de la pensée de Vygotsky sur les construits culturels page 6 (modèle constructiviste).

menées des recherches dans un climat de coopération» (Crahay, 1999, p. 348). La méthode FLC (*Fostering Learners Community*) qu'ils proposent, vise à promouvoir des niveaux élevés de cognition et métacognition en encourageant les élèves à rechercher, partager, et produire ensemble un artefact faisant sens à ses yeux (apprentissage situé).

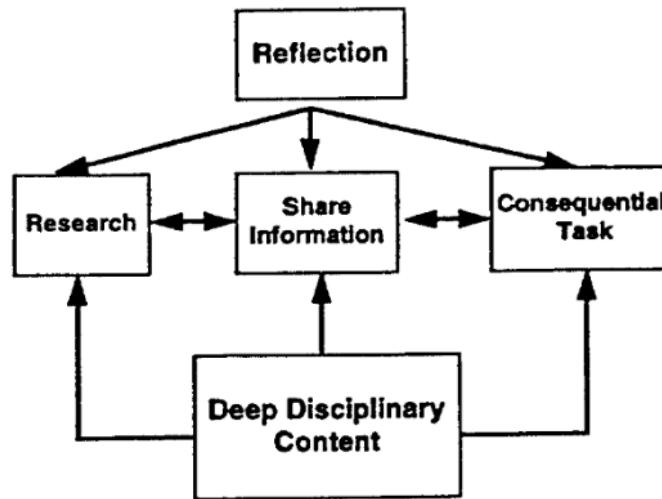


Figure 4: Base du système d'activité pour une pratique FLC. (Brown & Campione, 1996, p. 293)

En résumé, toute approche socioconstructiviste postule que l'acquisition de connaissances durables est favorisée par la prise en compte de l'environnement social dans lequel elle est située. Les apprentissages ont lieu via des processus interactionnels « individu-groupe », « groupe-environnement ». Le savoir n'est plus exclusivement transmis par un enseignant mais résulte de l'activité d'un groupe agissant sur un environnement spécialement aménagé par l'enseignant dans un cadre historiquement et culturellement située.

4.3 TICE et socioconstructivisme.

Les approches socioconstructiviste vont notamment (re)mettre en avant les pédagogies par projet incluant une dynamique collective. La connaissance est alors moins un savoir que l'on accumule qu'une ressource dont la maîtrise sert l'avancement d'un projet commun. Ces pédagogies par projet vont rapidement être soutenues par les TIC. En effet ces dernières offrent des potentialités de design, de production collaborative, de logistique et de communication favorable à ce type d'approche, pour peu qu'elles soient utilisées dans une véritable vision pédagogique. Comme le constate Dillenbourg & Jermann (2002) « certains usages actuels ne correspondent à aucune évolution pédagogique : mettre ses notes de cours sur Internet au lieu de les imprimer est une évolution logistique davantage que pédagogique ». De même Peraya (2002) nous rappelle que « les TIC ne sont que des outils au service d'un projet pédagogique. L'innovation se situe au niveau des pratiques pédagogiques ». Ainsi il apparaît comme erroné de qualifier un dispositif d'enseignement de « socioconstructiviste » par le simple fait qu'il incorpore un outil de communication tel qu'un chat ou un forum permettant le dialogue entre apprenant. Enfin, si les pédagogies socioconstructivistes bénéficient d'une popularité croissante, leur mise en œuvre peut donc s'avérer complexe pour deux raisons principales à nos yeux :

- *Les apprentissages ne vont pas de soi et sont dépendants de la rigueur avec laquelle l'enseignant a planifié et orchestré les activités.* Si la dynamique de construction collective des savoirs favorise l'acquisition en profondeur des connaissances, cela s'opère au prix d'une scénarisation précise des activités, de rétroactions importantes et d'une gestion efficace des dynamiques de groupe.
- *Les pédagogies socioconstructivistes nécessitent souvent, capacités communicationnelles et autonomie de la part des apprenants.* Hors ces capacités sont parfois difficiles à apprécier pour l'enseignant, voir différemment grandement au sein d'un même groupe.

CONCLUSION

À travers ce panorama des principaux courants théoriques, se dessine progressivement les tendances de notre vision actuelle de l'apprentissage. Une vision qui sans renier pour autant les apports de chacune des conceptions reflète une évolution allant d'une conception de l'apprentissage individuel à une conception de l'apprentissage en communauté, d'une approche basée sur les contenus à une approche axée sur les processus, de contenus médiatiques statiques et expositifs à des contenus dynamiques et interactifs, de ressources isolées à leur utilisation intégrée. Ces évolutions placent définitivement l'apprenant en tant qu'acteur au sein d'un dispositif techno-pédagogique complexe dont l'enseignant est l'architecte. Il peut être alors nécessaire pour un enseignant de se livrer à une activité de « scénarisation » afin de planifier son enseignement mais aussi l'activité des apprenants au sein de ce que l'on pourrait qualifier une petite « société d'apprentissage ».

Passant de l'approche transmissive au behaviorisme, cognitivisme et constructivisme, ce ne sont plus uniquement les contenus qui sont organisés, structurés, scénarisés mais l'ensemble du dispositif de formation, incluant l'activité des acteurs. Le scénario traduit alors une certaine conception de l'enseignement, une certaine approche pédagogique. Comme le constate Dessus (2006) ou Henri, Compte et Charlier (2007) les scénarios pédagogiques suivent l'évolution des théories de l'apprentissage et se distinguent aujourd'hui par leurs capacités à prendre en compte le contexte, l'histoire, l'activité, voir le projet de l'apprenant pour favoriser des apprentissages signifiants.

C'est dans ce contexte que nous présentons ce mémoire où nous abordons la question de la scénarisation du point de vue des technologies éducatives. Nous nous focalisons sur la manière dont les technologies peuvent aider les enseignant à modéliser et formaliser leurs intentions pédagogiques et articuler les environnements informatiques autour de leurs scénarios. Dans notre approche, les TICE visent à *soutenir* les *processus* d'apprentissage. « *Soutenir* » car les technologies, tout comme les formateurs, ne pourront jamais que créer les conditions propices à l'apparition d'un apprentissage. « *Processus* » car « apprendre » est un processus cognitif complexe émergeant au cours d'activités de découverte, de construction, de partage, de discussion, de confrontation, d'intégration, de renforcement... La scénarisation a dès lors pour objectif de structurer et organiser ces différentes activités en vue de provoquer des apprentissages identifiés.

Au chapitre suivant nous proposerons notre vision du concept de scénarisation avec les technologies puis présenterons comment elle est abordée actuellement en ingénierie pédagogique.

CHAPITRE II. SCÉNARISATION DE L’ENSEIGNEMENT

Une grande partie du travail des enseignants est réalisée en dehors de leur classe. Cette remarque est à la fois banale (on connaît le temps qu'ils passent à corriger des copies), et nécessaire (on leur fait souvent remarquer qu'ils ont trop de vacances). Toutefois, le travail de préparation de leur enseignement a longtemps été occulté, même par la recherche en éducation, qui ne s'y est intéressée que depuis une trentaine d'années. Ce gain d'intérêt révèle bien que la société et la recherche en éducation tend à considérer l'enseignant comme un professionnel : qui, en effet, dirait d'un avocat qu'il ne travaille que lorsqu'il plaide dans un tribunal ?

[Dessus, 2006, p.4]

INTRODUCTION

Les activités de scénarisation ont d'une certaine manière toujours été au cœur de l'activité d'enseignement. C'est l'acceptation du terme et le sens que l'on accorde à cette notion qui ont, quant à eux, évolué au cours du temps. Si l'expression « scénarisation pédagogique » est apparue après les années 60 avec l'intégration de l'audiovisuelle comme moyen pédagogique en classe (Henri, Compte et Charlier, 2007, p.16), les activités de planification de l'enseignement, le découpage et l'organisation du contenu d'apprentissage ont toujours été indissociables de la pratique des enseignants. Cependant l'intérêt pour les théories constructivistes de l'apprentissage, le développement des pédagogies actives, et la montée croissante des TICE dans l'enseignement a amené à scénariser l'enseignement de manière plus globale et structurée.

Dans un premier temps nous définirons brièvement l'Instructional Design dont nous considérons les modèles comme soutien à la conception de scénarios. Nous présenterons ensuite une définition selon Pernin et Lejeune sur laquelle nous appuieront grandement notre vision de la scénarisation. Enfin, nous présenterons la Théorie de l'Activité que nous prenons comme un cadre conceptuel important pour la modélisation pédagogique avec BPMN proposée dans les chapitres suivants de ce mémoire. Ces trois présentations nous permettront d'introduire notre conception de la scénarisation sous l'angle d'un « workflow » orchestrant le processus d'apprentissage dans un environnement techno-pédagogique.

1. DE L’INSTRUCTIONAL DESIGN À LA SCÉNARISATION PÉDAGOGIQUE.

1.1 Définition de l'ID.

L'Instructional Design¹² peut-être défini avec Dessus (2006) comme un champ de recherche s'intéressant « aux aspects prescriptifs de la conception de l'enseignement, c'est-à-dire à la manière dont on peut organiser ce dernier pour qu'il soit le plus efficace possible » (in abstract). Pour Reigeluth (1999) L'ID est avant tout « une orientation, un guidage, explicite sur comment mieux aider les gens à apprendre et à se développer » (p.5). Siemens (2002) quant à lui, propose parmi d'autres la définition suivante :

¹² Comme le remarque certains auteurs (Paquette, 2005, p.110 ; Dessus, 2006, p.4). Il existe un flou sémantique et terminologique autour de la notion d'Instructional Design aussi bien dans la littérature anglophone que francophone. C'est pourquoi nous garderons la terminologie « Instructional Design » sans traduction tout au long de ce mémoire.

« *L’Instructional Design est l’art et la science de créer un environnement matériel, didactique et pédagogique qui permettra à l’apprenant de passer d’un état où il n’est pas en mesure d’accomplir certaines tâches à l’état d’être en mesure d’accomplir ces tâches. L’Instructional Design s’appuie sur des recherches théoriques et pratiques dans les domaines de la cognition, psychologie de l’éducation, et la résolution de problèmes.* »

Quelle que soit la définition choisie comprenons que l’ID n’est sûrement pas une théorie de l’apprentissage, mais une « discipline » transversale qui définit des modèles (au sens de méthodes ou procédures) liant les théories de l’apprentissage aux pratiques éducatives¹³. Les théories de l’ID ne sont pas des *descriptions* des processus d’apprentissage mais des *prescriptions* favorisant l’apprentissage (Paquette, 2005). Originellement lié aux programmes bémoriste d’entraînement militaire de la seconde guerre mondiale, l’ID a progressivement évolué pour s’adapter aux théories de l’apprentissage dominantes de chaque époque (Dessus, 2006). S’il est ainsi possible de classer les modèles d’ID selon la perspective behavioriste, cognitiviste et constructiviste sous-jacente à leur développement (Van Merriënboer & Kirschner, 2001 ; Dessus, 2006), constatons que l’Instructional Design reste fondamentalement marqué par ces origines. De ce fait la planification systématique, l’approche par procédure, la recherche d’efficacité sont des constantes de l’ID. Des constantes qui, associées au développement de l’informatique dans l’enseignement, vont naturellement rapprocher l’ID des méthodes d’ingénierie. Mettant en lumière ce lien¹⁴, Paquette (2005) propose une définition pertinente à nos yeux de L’ID (qu’il traduit par *design pédagogique* en français) :

« *Le design pédagogique se présente aujourd’hui comme l’ensemble des théories et des modèles permettant de comprendre, d’améliorer et d’appliquer des méthodes d’enseignement favorisant l’apprentissage. Par rapport aux théories élaborées en psychologie de l’apprentissage, le design pédagogique peut être vu comme une forme d’ingénierie visant à améliorer les pratiques éducatives. Sa relation à la pédagogie est analogue à celle des méthodes de l’ingénieur par rapport aux sciences physiques, ou à celle de la médecine par rapport aux sciences de la vie.* » (p.111)

Le concept de scénarisation va rapidement trouver écho dans le courant de l’Instructional Design. En effet, d’une manière générale nous pouvons dire qu’un modèle d’ID raconte comment concevoir des scénarios pédagogiques appropriés pour atteindre certains objectifs d’enseignement. En ce sens, nous pouvons considérer qu’un scénario pédagogique concrétise l’action de L’ID.

1.2 Définition de la scénarisation pédagogique.

Si comme nous l’avons évoqué la structuration, l’organisation, la planification ont toujours été au cœur de l’activité d’enseignement pourquoi un tel engouement pour le terme *scénarisation* ? La raison majeure, selon nous, tient dans le fait qu’un *scénario* sous-entend les notions d’*acteur*, de *rôle* et d’*orchestration*. Il ne s’agit plus d’organiser et structurer des contenus de connaissance mais l’activité de l’apprenant, le rôle de chacun des acteurs de la formation et le rapport aux outils médiatiques. Ainsi, l’activité d’enseignement se rapproche effectivement de celle d’un scénariste et d’un metteur en scène, d’un compositeur et d’un chef d’orchestre. Nous retrouvons ci-dessous la définition de Pernin et Lejeune (2004) :

¹³ Pour reprendre la formulation bien connue de John Dewey (1900) évoquant un besoin important de créer « *a linking science between learning theory and educational practice* ». 150 ans plus tard, il est possible de percevoir l’ID comme cette « science intermédiaire » faisant défaut de son temps.

¹⁴ Les liens qu’entretiennent L’ID et l’ingénierie sont indéniablement étroits, cependant il nous paraît nécessaire de rappeler de ne pas les confondre pour autant. Au cours de nos lectures il n’était en effet pas rare d’observer l’utilisation du terme « ingénierie pédagogique » pour aborder l’ID. Si cette terminologie relève la plupart du temps d’une traduction maladroite de la part de l’auteur, elle nous a aussi semblé, dans certain cas, relevée d’une réelle confusion sémantique. Nous aborderons l’ingénierie pédagogique au point 3 de ce chapitre.

« Le scénario d'apprentissage représente la description, effectuée *a priori* ou *a posteriori*, du déroulement d'une situation d'apprentissage ou unité d'apprentissage visant l'appropriation d'un ensemble précis de connaissances, en précisant les rôles, les activités ainsi que les ressources de manipulation de connaissances, outils, services et résultats associés à la mise en œuvre des activités. » (p. 5)

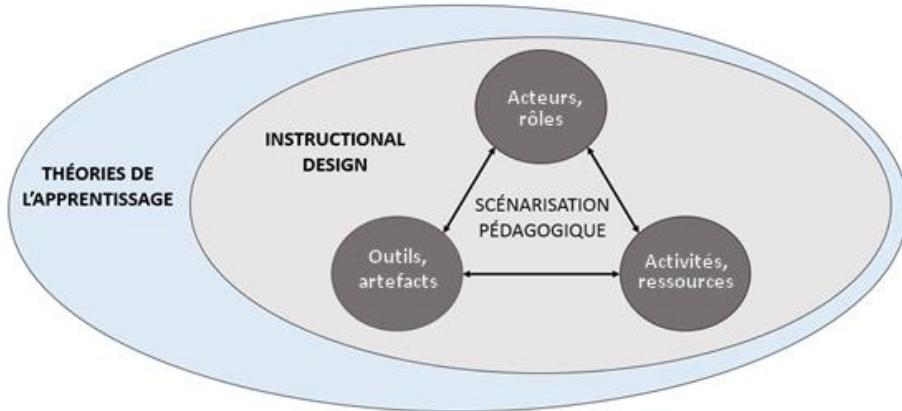


Figure 5: Représentation schématique de la scénarisation pédagogique.

Scénariser est donc, de manière générale, un processus de modélisation de la situation d'apprentissage. Cependant constatons que cette modélisation recouvre des aspects divers au sein desquels un formateur peut se situer à différents niveaux de granularité avec un degré de formalisation plus ou moins grand. Afin d'aborder la diversité des scénarios possiblement conçus, Pernin et Lejeune (*ibid.*) proposent 6 critères permettant d'effectuer une taxonomie des scénarios d'apprentissage. Nous présentons ci-dessous un résumé de chacun des critères :

- *Finalité d'un scénario.* Les auteurs effectuent une distinction entre un scénario réalisé *a priori* et *a posteriori*. Dans le premier le scénario est *prédictif* en vue de la mise en place d'une situation d'apprentissage. Dans le second cas le scénario est *descriptif*, relatant une situation effective en vue de la reproduire. La modélisation de ce second cas intègre les traces et productions des acteurs lors du précédent déroulement.
- *Granularité d'un scénario.* Pernin et Lejeune distinguent ici 3 niveaux. Le *déroulement d'une activité élémentaire*, *l'enchaînement des activités élémentaires* et la *structuration pédagogique* se situant à un niveau plus macroscopique et décrivant la structuration des cours, modules etc.
- *Degré de contrainte d'un scénario.* Les auteurs se réfèrent ici au degré d'ouverture et d'adaptabilité plus ou moins grand d'un scénario par les acteurs amenés à le mettre en place. Plus un scénario a un degré de description fin moins il laisse de place à l'initiative des acteurs.
- *Degré de personnalisation d'un scénario.* Distinction entre un scénario *générique*, identique à chacune de ses exécutions, et *adaptatif* prévoyant des chemins exécutés de manière conditionnelle.
- *Degré de formalisation d'un scénario.* Pernin et Lejeune abordent ici les langages de modélisation pédagogique (LMP) dont nous reparlerons plus précisément au point 3 de ce chapitre. Un enseignant peut créer un scénario *informel* pour les besoins de son enseignement ou au contraire modéliser à travers un langage *formel* favorisant le partage, la réutilisation voir l'automatisation partielle ou totale du scénario à travers des environnements informatiques.
- *Degré de réification d'un scénario.* Un scénario peut être *abstrait* (peu ou pas de prise en compte du contexte d'exécution), ou *contextualisé* (très lié à l'environnement, pouvant même aller jusqu'à énoncer les noms de personnes réelles à l'intérieur du scénario).

Ces critères sont selon nous un bon outil pour appréhender la diversité des scénarios ou orienter la réflexion préparatoire à la conception.

Pour finir ajoutons à l'outil de taxonomie précédent, un autre aspect important pour aborder la diversité des scénarios : le cadrage théorique sous-jacent. Les théories de l'apprentissage sont et seront toujours un cadre à la conception de scénarios pédagogiques (voir figure 5). Soit de manière explicite, les auteurs y faisant directement référence, soit de manière implicite même aux yeux de l'auteur, qui sans en avoir réellement conscience retranscrit dans son scénario les conceptions dominantes sur l'apprentissage de son époque ou de son environnement culturelle.

2. THÉORIE DE L’ACTIVITÉ UN CADRE CONCEPTUEL

La théorie de l'activité (TA) est parfois citée dans quelques travaux autour de la scénarisation pédagogique et des langages de modélisation pédagogique (Taurisson, 2005 ; Bourguin & Derycke, 2005 ; Martel & al. 2006 ;). Elle y est alors vue en tant que cadre au développement de pédagogies basées sur l'activité, la signification des actions d'apprentissage et l'usage des outils pédagogiques (Ex : Pédagogie de l'activité – Taurisson, 2005 ; Learning by expanding - Engeström, 1987).

Dans ce mémoire, la TA nous permet de conceptualiser l'activité au sein d'un dispositif pédagogique au travers de chaînes d'actions collectives, orientées vers un objectif socialement partagé et en interaction avec les acteurs et l'environnement. Elle nous permet d'introduire la notion de « workflow » en considérant l'apprentissage comme un *processus non séquentiel* et instrumenté, ancré dans une dynamique communautaire. Initialement issue des travaux de l'école historico-culturelle soviétique fondée par Vygotski et ses disciples (principalement Leontiev), elle sera reprise par l'école Scandinave avec notamment les travaux d'Engeström (1987). En abordant *la médiation par l'outil* (ou instrument) la TA offre un cadre théorique au domaine des TIC et, en prenant comme unité fondamentale *l'activité du sujet*, la TA s'enracine dans une vision (socio)constructiviste de l'apprentissage.

2.1 L’activité une entité composite.

Pour Leontiev (1979) les activités sont toujours en relation avec un but conscient, une motivation. Un but peut être subdivisé en un ensemble d'actions intermédiaires parfois socialement distribuées. Les activités de chacun forment alors un système. L'action d'un seul acteur ne peut être comprise qu'au regard de l'ensemble du système. Leontiev (cité par Kuutti, 1995) illustre ce principe avec l'activité de chasseurs primitifs qui se scindent en deux groupes pour attraper leurs proies, les chasseurs qui tueront l'animal et les rabatteurs qui l'orientent. Si l'on isole les actions de ces derniers et que nous les comparons au motif de chasse qui est de capturer un gibier pour sa viande et sa fourrure – les actions des rabatteurs sont totalement irrationnelles ; elles ne peuvent être comprises qu'en étant une partie d'un système plus vaste d'activités humaines. Ainsi, selon Leontiev, l'activité peut être décrite par une structure hiérarchique en 3 niveaux : l'activité, l'action, et l'opération. L'activité (niveau le plus haut) est réalisée via des chaînes d'actions (niveau intermédiaire) elles-mêmes réalisées au travers d'opérations (niveau le plus bas). Les actions peuvent être individuelles ou collectives, mais sont toujours dirigées vers un but conscient (intermédiaire par rapport à ce qui motive l'activité). Comme le montre l'exemple des chasseurs précédent il est difficile de comprendre une action hors de son contexte : l'activité. Cependant, une action peut appartenir à diverses activités et son interprétation dès lors diffère selon l'activité considérée.

Activity level	- Building a house	- Completing a software project	- Carrying out research into a topic
Action level	- Fixing the roofing - Transporting bricks by truck	- Programming a module - Arranging a meeting	- Searching for references - Participating in a conference - Writing a report
Operation level	- Hammering - Changing gears when driving	- Using operating system commands - Selecting appropriate programming language constructs	- Using logical syllogisms - Selecting appropriate wording

Figure 6: Exemple d'activité, action et opérations selon Kuutti (2005)

Au final l'ensemble des actions forment un plan calqué sur des modèles jugés efficaces pour atteindre l'objectif. Selon Bourguin & Derycke (2005) « *Plus le modèle mis en œuvre est fiable, plus l'action a des chances d'atteindre son but. Dans le cas contraire, le sujet entre dans une phase d'analyse et d'apprentissage au cours de laquelle il sera certainement amené à modifier le modèle défaillant et, ainsi, augmenter son expérience.* » (p. 7). Si la théorie de l'activité par son intérêt pour l'action de l'utilisateur est en lien étroit avec l'apprentissage, elle n'est cependant pas une théorie de l'apprentissage. La TA fournit plutôt un cadre pour le développement de dispositifs d'apprentissage en proposant de modéliser divers aspects de l'activité en un tout unifié.

2.2 Représentations triangulaire de l'activité.

Parmi les aspects d'une activité, trois éléments sont au fondement de la théorie de l'activité : Le sujet, l'outil et l'objet. **Le sujet** est en accord avec un objet (ou objectif). Son activité est orientée vers l'atteinte de cet objet (motivation objectivé). **L'outil** est matériel ou symbolique et médiatise l'activité. Il est à la fois favorable et limitant. En effet, un outil habile, donne le potentiel au sujet de réaliser une activité avec les compétences, l'expérience historiquement collecté et "cristallisé" dans l'usage de cet outil (Vygotski). Cependant il restreint également l'interaction au sein de l'activité à la perspective donnée par cet outil particulier – les autres options possibles peuvent rester « invisibles » aux yeux du sujet (d'après Kuutti, dans Nardi, 1996, p.25). **L'objet** oriente les actions de l'activité, il en est le moteur. L'objet est lié à des processus motivationnels pouvant se modifier au cours de l'activité lors de la confrontation « au faire », à la réalisation de la tâche. Pour Kuutti (ibid.) transformer l'objet en un *produit*, un *résultat*, motive l'existence de l'activité, donne sens à l'action.

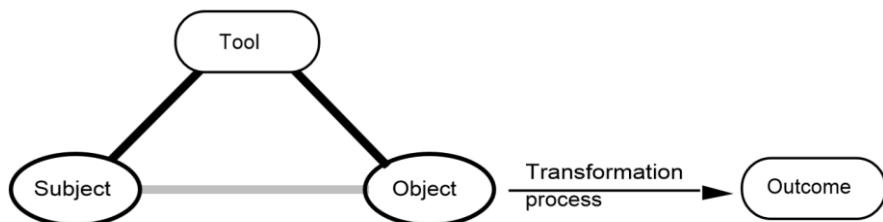


Figure 7: Structure de base de l'activité humaine médiatisée (Nardi, 1996, p.25).

Cette relation triangulaire de base fondée sur le concept de médiation tel que théorisé par l'école historico-culturelle russe, est essentiellement centrée sur l'individu et va progressivement être étendue à un modèle systémique plus complexe comprenant l'environnement. Généralisant les concepts développés par Vygotski et Leontiev, Engeström propose une extension du triangle conçue pour être un cadre au déploiement et à l'analyse d'une activité : « *le plus petit environnement indispensable pour donner du sens à ce que l'on fait* » (Taurisson, 2005).

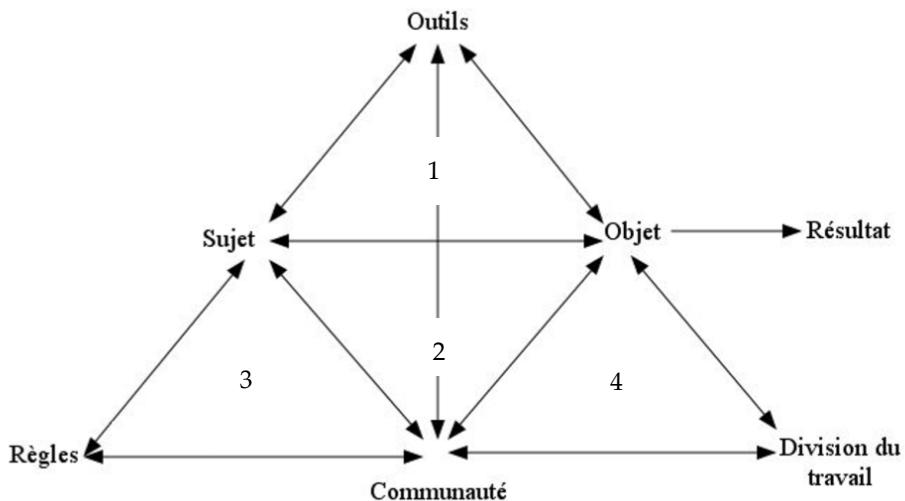


Figure 8: Structure d'une activité d'après Engeström (1987)

Dans ce modèle, le sujet s'intègre dans une communauté interagissant avec le même objet (ex : corps hospitalier ayant pour objet les patients à soigner). La communauté contribue aussi à sa production ou à sa transformation en résultat grâce à la présence de règles (explicites ou non) régulant les interactions sujet/communauté et la division du travail au sein de la communauté. L'apparente simplicité du modèle peut être trompeuse. Une analyse plus avancée révèle un modèle à la systémique complexe permettant d'aborder l'activité selon un plan historico-temporel et spatial-social. Sans entrer dans les détails, ce modèle permet d'aborder dans un premier temps l'analyse d'une activité selon 4 types de médiation. Le triangle principal est formé d'un minimum de 4 triangles. La relation du sujet à l'objet est médiée par les outils (triangle 1) et par la communauté (triangle 2). La relation sujet-communauté est médiée par un ensemble de règles (3) tandis que la relation entre l'objet et la communauté est médiée par une division du travail spécifique. L'investigation des dynamiques relationnelles au sein de ces 4 triangles permet une première approche quant à l'étude d'une activité mais le modèle est plus riche que cela et propose d'autres perspectives. Ainsi le triangle « sujet-outil-communauté » nous invite à aborder l'analyse d'une activité en nous penchant sur le processus historico-social de construction des usages d'outil matériel ou symbolique. Le triangle « outil-communauté-objet » quant à lui, permet d'analyser les objectifs à atteindre par rapport aux outils disponibles et à l'usage que la communauté en fait. Pour utiliser ce modèle, il s'agit donc de définir dans un premier temps chacun des différents pôles (avec le degré de détails nécessaire) et ensuite d'analyser les interactions de manière systématique.

2.3 TA et scénarisation pédagogique.

Dans le champ de l'éducation, la modélisation d'un scénario peut tout à fait ne décrire qu'une seule activité - ce sera même souvent le cas lorsque l'on vise un niveau de détail suffisant pour l'automatisation de certaines procédures. Une activité peut donc être complexe en étant l'aboutissement d'une succession d'actions et d'opérations (routines souvent scriptées et effectuées par le dispositif informatique dans notre cas). Une activité a un objet vers lequel vont tendre chacune des actions de l'apprenant via l'ensemble des outils et artefacts mis à sa disposition par l'enseignant et le groupe, ou tout simplement choisi par l'apprenant lui-même. La plupart du temps en éducation, l'objet de l'activité sera transformé en un résultat tangible, (typiquement une production écrite, la fabrication d'un produit) et/ou dans le cas contraire laissera un certain nombre de traces - numériques ou non - témoignant des actions de l'élèves. Là où réside la difficulté, est selon nous dans la différenciation entre une activité et une action. De manière générale en éducation, nous parlerons d'activité chaque fois qu'il y a production matériel ou immatériel d'un résultat perceptible témoignant l'atteinte de l'objectif. Comme nous l'avons vu précédemment une action peut appartenir à diverses activités et son interprétation dès lors diffère selon l'activité considérée. Pour illustrer notre propos prenons les situations suivantes.

- *Cours universitaire avec des étudiants habitués à utiliser une plateforme d'apprentissage :* Lire un document est pour eux une action, une tâche prescrite par l'enseignant comme préparation dans le cadre d'une activité plus vaste. Cette action met en œuvre certaines opérations souvent routinières de la part de l'élève (allumer son ordinateur, ouvrir son navigateur, cliquer sur le lien du document, commencer par lire l'introduction et la conclusion puis parcourir le document, etc.). Réussir à ouvrir le navigateur n'a, en aucun moment, été perçu comme un but intermédiaire par l'apprenant.
- *Cours d'initiation à l'utilisation de l'informatique pour personnes du troisième âge :* Ouvrir ce même document est ici un objectif en soi, une activité. L'apparition du texte à l'écran apparaîtra ici comme le résultat des actions entreprises par le sujet en vue de cette finalité. Chacune des opérations de l'expert peut dès lors devenir une action pour le novice faisant intervenir un certain nombre d'opérations nouvelles (corriger erreur, chercher information, demander de l'aide etc.). L'ouverture du navigateur est perçue comme une première étape, l'atteinte d'un sous-but pour le sujet.

Dans les deux cas précédents la tâche est en partie similaire (tous deux ouvrent un document sur le web). Mais nous avons affaire à des sujets différents dans des activités distinctes. Dans le premier cas l'objet est la réalisation d'une future production écrite. Dans le second cas, l'objet est l'ouverture d'un document. L'ouverture du document est une simple opération dans le premier cas, alors qu'elle représente la finalité d'une activité dans le second.

CONCLUSION

Si les modèles d'ID se proposent d'articuler les théories et les pratiques éducatives, la TA offre quant à elle un cadre conceptuel pour comprendre l'activité.

Dans ce mémoire, nous positionnons l'activité comme l'unité fondamentale de la conception d'un scénario pédagogique dont la dynamique et l'enchaînement des actions individuelles et collectives découlent d'une approche pédagogique orientée vers un objectif d'apprentissage, une compétence à développer chez l'apprenant. L'approche que nous proposons positionne l'apprentissage dans un flux d'actions sociales et médiatisés dont la dynamique est l'essence du système. La scénarisation pédagogique devrait dès lors pouvoir décrire un « processus », un « flux », un « workflow ». Notre approche centrée processus ne peut être soutenue ni par des technologies simplement orientées sur la description standardisée de contenu d'apprentissage, ni par des technologies concevant l'activité comme une séquence linéaire de tâches à effectuer. Au contraire, les technologies devraient, selon nous, permettre au concepteur pédagogique de modéliser et partager des scénarios rendant compte des enchaînements complexes et des cheminements divers qu'une situation d'apprentissage comporte, afin de retranscrire au mieux la dynamique des processus d'acquisition de connaissances.

Modéliser et implémenter de telles intentions pédagogiques devient dès lors une tâche complexe pour un concepteur et un défi technologique pour les outils de support et d'exécution. La résolution de cet enjeu nécessite le soutien de méthodes normalisées d'ingénierie pédagogique. Nous verrons au chapitre suivant comment ce domaine de recherche aborde cette problématique à travers, notamment, les travaux sur les langages de modélisation pédagogique et la conception des EIAH dirigée par les modèles.

CHAPITRE III. INGÉNIERIE PÉDAGOGIQUE

« Historiquement, si un très grand nombre d'enseignants ont pris le parti de ne pas s'investir dans la scénarisation et la production de ressources audiovisuelles et de demeurer simples utilisateurs de produits faits par d'autres, ce choix n'est plus possible avec les environnements d'apprentissage informatisés puisqu'il s'agit « d'enseigner en ligne ». Les enseignements ne peuvent être pensés et conçus par d'autres que les enseignants. Ces derniers doivent être les concepteurs de l'environnement d'apprentissage qu'ils utiliseront pour accomplir leur tâche d'enseignant. »

[Henri, Compte & Charlier, 2007]

INTRODUCTION

L'apparition des technologies a introduit une nouvelle évolution de l'ID, celle de l'ingénierie pédagogique. Cette dernière peut être définie selon Paquette (2005) par « *une méthodologie soutenant l'analyse, la conception, la réalisation et la planification de l'utilisation des systèmes d'apprentissage, intégrant les concepts, les processus et les principes du design pédagogique¹⁵, de l'ingénierie des systèmes d'information et de l'ingénierie cognitive* » (p. 106). L'introduction toujours plus grande des TICE dans l'enseignement transforme profondément l'environnement pédagogique. Les dispositifs de formation articulent aujourd'hui à des degrés divers des sessions de formation en présentiel et à distance soutenues par un environnement médiatisé (techno-pédagogique). Dès lors, le scénario créé par l'enseignant n'est plus seulement un outil pour guider son travail et celui de l'apprenant, il représente au contraire la pierre angulaire de l'environnement informatisé dont la conception « *obéit aux exigences de la conduite d'un projet informatique* » (Ernst, 2006, p.1). La complexité croissante des systèmes d'information peut amener les concepteurs d'environnements médiatisés d'apprentissage à utiliser des méthodes normalisées d'ingénierie venant soutenir leur activité.

Afin d'aborder la complexité d'une situation d'apprentissage médiée par ordinateur, la modélisation (au sens large) est importante. L'ingénierie pédagogique va s'intéresser à différentes méthodes et technologies permettant de modéliser et rendre interopérable divers aspects de l'enseignement et de l'apprentissage (contenus, activités de l'apprenant, interactions sociale, parcours d'apprentissage etc.). Nous distinguerons ici deux approches. Une première approche centrée sur la définition de standards permettant d'opérationnaliser et d'échanger des contenus, voire de manière plus récente, des activités d'apprentissage conçues autour de la notion de scénario (section 1). Une seconde approche issue des recherches en CSCL (computer-supported collaborative learning) abordant la modélisation des activités d'apprentissage collaboratives sous la forme de scripts structurant diverses interactions (section 2). Les deux sections précédentes nous permettront d'introduire la question des langages de modélisation pédagogique (Section 3) avec une centration particulière sur les langages graphiques. Enfin nous aborderons la question des acteurs de la formation face à l'ingénierie (section 4) et l'ingénierie des EIAH (section 5).

1. STANDARDISER LES OBJETS D'APPRENTISSAGE

1.1 D'une centration sur le contenu, à une centration sur l'activité.

Le concept « d'objet d'apprentissage » est en ingénierie une notion « vague » référant à un construit pédagogique, une ressource, un composant d'enseignement manipulable et échangeable. Construire ces objets d'apprentissage avec l'informatique comporte souvent un coût en ressources temporelles ou financières

¹⁵ Le design pédagogique est la traduction française utilisée par Paquette pour « Instructional Design »

important. Dès lors, apparaît la volonté d'amortir ce coût sur le nombre et la durée d'usage ou sur la réutilisation potentielle des objets d'apprentissage. L'ingénierie pédagogique va rapidement se baser sur un ensemble de normes et de standards afin de faciliter l'interopérabilité, la diffusion, le partage et la réutilisation de ces objets d'apprentissages. De manière générale, ces spécifications ont pour objectif de décrire de façon standard « toute ressource numérique, reproductible et adressable, utilisée pour réaliser des activités d'apprentissage ou d'encadrement de l'apprentissage et rendue accessible à d'autres pour leur utilisation » (Koper 2003, citée par Pernin & Leujeune, 2004, p.2).

L'approche « E-learning classique» s'est dans un premier temps conçue comme un ensemble de « briques et agrégat de contenu» indépendantes et réutilisable que l'on assemble pour construire une formation. C'est le développement des « content packages » décrivant, structurant, séquençant de manière simple des ressources et permettant leurs partage et exécution dans des environnements conformes. IMS Content Packaging et ses extensions SCORM sont les standards les plus connus de cette approche.

Comme le constate Pernin et Lejeune (2004) l'ingénierie pédagogique va progressivement s'intéresser à une manière différente de percevoir les objets d'apprentissages recentrée sur la notion d'activité. Cette nouvelle approche redéfinie les objets d'apprentissage non seulement comme des unités de contenus, mais comme un ensemble d'actions autour d'un scénario. Ces activités sont réalisées par des acteurs tenant un rôle précis et se déroulent dans un environnement caractérisé par un ensemble de services, d'outils et de ressources pédagogiques. L'approche activité va donc remettre le scénario pédagogique au centre de l'ingénierie de formation. Pour formaliser et opérationnaliser ces scénarios la recherche se tourne vers le développement d'autres spécifications. Ainsi se développe des normes telles que IMS Learning Design et d'une manière plus générale un domaine de recherche centré sur les langages de modélisation pédagogique (LMP).

1.2- IMS CP, LOM, SCORM.

IMS Content Packaging est un ensemble normalisé de structures pour « l'emballage » d'objet d'apprentissage. Ces principaux objectifs sont de permettre la transmission d'un contenu d'une plateforme vers une autre, faciliter l'importation / exportation de contenu dans un répertoire ou une plateforme e-learning, permettre l'assemblage de contenus simples dans un paquet de distribution, permettre la production avec des outils d'édition indépendants des plateformes. Un IMS content Pack se réfère toujours à une archive .zip (PIF - Package Interchange File) contenant des ressources variées (page HTML, images, animations Flash, documents, etc.) et un manifeste sous la forme d'un fichier XML. Ce dernier constitue le cœur d'IMS-CP et contient :

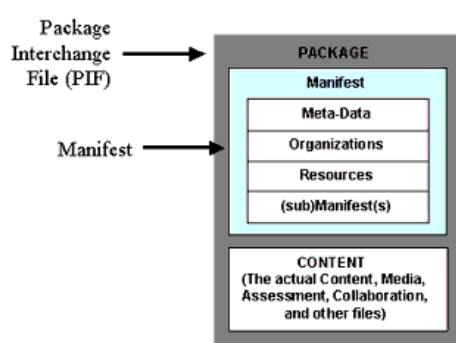


Figure 9: Architecture du Package Interchange File (PIF)

1. Une section Meta-Data qui décrit le package IMS dans son entier. Les métadonnées sont généralement décrites via des schémas standards de description. Le standard générique Dublin Core ou le standard plus spécifique à l'éducation LOM (Learning Object Metadata) sont les exemples les plus utilisés.
2. Une section de ressources qui liste les éléments dans l'archive et des ressources disponibles via un URL sur le Web (ou en intranet)
3. Une section d'organisation qui décrit la structure simple des ressources (« En arbre » à la manière d'un menu). Cependant, en recentrant la définition des objets d'apprentissage sur la notion d'activité cette section va pouvoir intégrer d'autre spécification permettant une scénarisation plus évoluée (IMS Simple Sequencing, IMS Learning Design...)

Dans une perspective d'opérationnalisation, le consortium ADL (Advanced Distributed Learning) va développer le standard SCORM reposant sur IMS CP et LOM. De manière schématique SCORM est un standard qui définit comment créer des contenus e-learning et la manière dont ils interagissent avec une plateforme de gestion de contenus d'apprentissage (LMS). SCORM se compose :

- D'un modèle d'agrégation de contenu (Content Aggregation Model), vu comme une extension d'IMS, fournissant un guide pour l'identification des ressources de base et leur agrégation dans un contenu structuré de formation. Ce modèle prend en compte les informations du LOM;
- D'un environnement d'exécution (Run-Time Environment) fournissant un guide pour l'exécution des contenus, la communication et le suivi des activités dans un environnement web. Selon Pernin (2003) cela « permet aux développeurs de Systèmes de Gestion de Formation de disposer d'un cadre pratique pour intégrer des objets pédagogiques ».

SCORM 1.2 fut la version référence, utilisant la spécification IMS Content Packaging 1.1.2 et son manifeste XML pour décrire le contenu et le parcours d'apprentissage. SCORM n'est plus maintenu ou soutenu par ADL mais reste très utilisé en e-learning. SCORM 2004 quant à lui est la dernière version du standard. Comme nous l'avons déjà énoncé une certaine tendance de l'ingénierie pédagogique recentre l'approche de conception des objets d'apprentissage sur la notion d'activité. SCORM 2004 s'intègre dans cette mouvance. En se basant sur IMS-CP 1.1.4 ce nouveau standard améliore la navigation (Bouton avant, arrière, stop...) et le séquencement des objets d'apprentissage (IMS Simple Sequencing). The IMS Simple Sequencing définit une méthode, à travers une syntaxe XML claire, pour représenter le comportement attendu des apprenants dans la structure selon une approche très behavioriste. Il permet d'adapter le parcours et de gérer plus précisément l'avancement et l'ordonnancement des contenus grâce aux objectifs d'apprentissage ou résultats aux tests en s'affranchissant des fonctionnalités du LMS¹⁶.

Poursuivant une approche centrée sur les processus d'apprentissage et l'activité de l'apprenant, l'IMS/GLC (Instructional Management Systems Global Learning Consortium) va s'intéresser à la formalisation et l'opérationnalisation d'objets d'apprentissage concrétisant les intentions pédagogiques autour d'un scénario.

1.3 – EML, IMS Learning Design.

IMS Learning Design est une spécification permettant de décrire de manière formelle des scénarios pédagogiques. IMS LD trouve sa source dans le langage EML¹⁷ développé à la fin des années 90 à l'Open University of the Netherlands (OUNL) sous l'impulsion des travaux de Koper. Ces derniers contribuèrent à remettre l'activité et l'orchestration d'activités au centre de l'ingénierie pédagogique. EML fut évalué par le Comité Européen de Normalisation (CEN), au cours d'une étude comparative de différents langages de modélisation pédagogique (CEN/ISSS WS/LT, 2002), comme répondant le mieux aux critères satisfaisant la définition d'un langage de modélisation pédagogique (Pernin & Lejeune, 2004). Il fut par la suite soumis au groupe de travail IMS Learning Design pour le développement de la spécification IMS-LD. C'est actuellement le standard le plus populaire pour décrire un « Learning Design » dans le domaine des langages de modélisation pédagogique (voir section 2). Selon (IMS Global Learning Consortium, 2002) *Learning Design* fait référence à «*une description d'un procédé permettant aux apprenants d'atteindre certains objectifs d'apprentissage par la réalisation de certaines activités pédagogiques dans un certain ordre dans le cadre d'un certain environnement d'apprentissage. Le Learning Design est basé sur les principes pédagogiques du concepteur, un domaine spécifique et un ensemble de variables contextuelles* ». De manière générale, IMS-LD permet de représenter et d'encoder des structures d'apprentissages

¹⁶ En pratique cependant notre expérience avec SCORM 2004 nous permet d'émettre quelques réserves. Les LMS gèrent encore imparfaitement ce standard ce qui oblige parfois le concepteur à exporter/importer en SCORM 1.2 et gérer le séquencement via les fonctionnalités du LMS (quand celles-ci le permettent).

¹⁷ Ne pas le confondre avec l'utilisation générique de l'acronyme EML (« Educational Modelling Language »). On le retrouve parfois dans la littérature sous l'appellation « OUNL-EML » pour éviter cette confusion.

pour des apprenants à la fois seuls et en groupe, rassemblés par rôles. On peut modéliser un plan de cours en IMS-LD, en définissant des rôles, des activités d'apprentissage, des services et plusieurs autres éléments et en construisant des unités d'apprentissage. Le plan de cours est modélisé et construit avec des ressources assemblées dans un dossier Zip compressé puis mis en route par un exécutable (« Player »).

IMS-LD est avant tout un *langage de description*. Il ne traite pas des processus d'instanciation et de mise en exploitation, lesquels sont sous la responsabilité de l'environnement d'exploitation : « *la portée du langage se limite à l'expression d'un modèle d'unité d'apprentissage (par exemple un cours), sous une forme structurée selon une notation XML. Ce cours pourra alors être diffusé en ligne par une plateforme compatible avec le standard après les étapes nécessaires pour sa mise en exploitation* » (Lejeune, 2004, p.415). De plus, il faut comprendre qu'IMS-LD est une couche d'intégration de plusieurs spécifications existantes. En effet l'IMS/GLC a, depuis 1997, développé un grand nombre de spécifications techniques. Ces spécifications sont utilisées à l'échelle internationale pour l'interopérabilité et l'implémentation de produits éducatifs. Chaque spécification représente un espace de nom XML et établit des relations de communication avec les autres spécifications IMS. À titre d'illustration, Lejeune (*ibid.*) répertorie les spécifications IMS suivantes :

- indexation des ressources d'apprentissage (IMS Meta-Data)
- profil de l'apprenant (IMS Learner Information Package)
- description des compétences (IMS Reusable Definition of Competency or Educational Objectives)
- interrelations au sein d'une organisation (IMS Enterprise)
- interopérabilité des systèmes et des environnements (IMS Content Packaging)
- évaluation des connaissances et des compétences (IMS Question & Test Interoperability)
- métamodélisation d'une unité d'apprentissage (IMS Learning Design)
- interopérabilité des banques de ressources pédagogiques (IMS Shareable State Persistence).

Ainsi dans IMS-LD, l'extension par l'utilisation ou l'inclusion d'autres spécifications IMS n'est pas obligatoire mais intervient très souvent à plusieurs niveaux dans la modélisation d'une unité d'apprentissage (*ibid.*).

Nous proposons en annexe 2 une description plus détaillée pour comprendre ce standard. Nous présentons le modèle structurel d'une unité d'apprentissage ainsi que les trois niveaux de conception (A, B et C) de la spécification.

2. SCRIPT CSCL

2.1 Computer-Supported Collaborative Learning.

De manière générale la recherche en CSCL (Computer-Supported Collaborative Learning) « *se focalise sur la manière dont l'apprentissage collaboratif assisté par la technologie peut améliorer l'interaction entre pairs et le travail en groupe, et comment la collaboration et la technologie facilitent le partage la distribution des connaissances et l'expertise entre les membres de la communauté.* » (Lipponen, 2002).

La recherche en CSCL puise ses fondements dans divers courants de pensées, domaines de recherche et théories tels que la recherche CSCW (Computer-Supported Cooperative Work) et CMC (Computer-mediated communication), les différentes approches socioconstructiviste (interactionnistes, socio-culturelles), la cognition distribuée, l'apprentissage situé, la Théorie de l'Activité. Selon, Sthal (2002) il est possible de définir 4 thèmes d'importance en CSCL :

1. La construction de connaissance en collaboration
2. Les perspectives individu/groupe
3. La médiation par les artefacts
4. L'analyse d'interaction.

La recherche en CSCL considère donc la communication comme au cœur de l'activité humaine et plus particulièrement de l'apprentissage. L'apprentissage collaboratif ne va pas de soi et dépend de processus interactionnels et cognitifs complexes qui se doivent d'être structurés et organisés en vue de provoquer des apprentissages au cours du processus.

2.2 Notion de script.

La recherche en CSCL va aborder l'ingénierie pédagogique par la formalisation de l'organisation des situations d'enseignement/apprentissage collaboratives sous formes de schémas et routines d'interactions: les scripts CSCL. Ces scripts sont un élément central de la modélisation pédagogique en CSCL, ils s'apparentent à « des scénarios définis comme une séquence de phases durant lesquelles les rôles sont distribués entre les apprenants » (Dessus, Pernin & Leujeune, 2007). Ces scripts sont documentés afin de rendre la collaboration réellement productive (taille de groupe, mode de constitution) et peuvent être considérés comme « recettes fournies aux concepteurs » (*ibid.*). Selon eux :

"Les scripts sont des méthodes structurant l'apprentissage collaboratif avec l'attente qu'une collaboration structurée est plus efficace qu'une collaboration libre. Ils favorisent l'émergence d'interactions productives de connaissances telles que l'argumentation et l'explication. Les mécanismes qui favorisent ces interactions sont exprimées par un principe générique: Un script « sépare » le travail d'équipes au regard des connaissances ou des rôles et cette séparation conduit les membres des groupes à s'engager dans des interactions pour compléter la tâche malgré la scission. La nature de la « scission » détermine les types d'interactions qui seront nécessaires pour s'engager à la surmonter. Par conséquent, ce principe est appelé "split where interactions should happen" (SWISH)."

Dillenbourg and Hong (2008) distinguent deux sortes de script :

1. Les micro-scripts décrivent des modèles discursifs la plupart du temps de type argumentation/débat que les élèves sont sensés adopter et progressivement internaliser.
2. Les Macro-scripts sont des modèles pédagogiques décrivant des séquences d'activités que les groupes doivent accomplir.

Les Macro-script sont assez semblables par certains aspects à l'approche adoptée par IMS-LD présentée précédemment. Ce dernier présente cependant un support insuffisant au regard de l'approche CSCL pour la description de d'interactions collaboratives synchrones. IMS à un profil plutôt « coopératif » orienté sur la personnalisation des parcours apprenant et le séquencement des tâches asynchrones.

Selon Kobbe (2006) les scripts CSCL offrent l'avantage de réduire l'effort de coordination à la fois du côté des enseignants et du côté des étudiants : « *Les ordinateurs peuvent, par exemple, garder la trace de la position des étudiants au sein de la séquence de script, alerter ou inciter les élèves à s'engager dans des activités spécifiques, et fournir des informations et des ressources supplémentaires en cas de besoin. De plus, une fois programmés, les scripts peuvent être réutilisés autant de fois et avec autant d'étudiants souhaités.* ». Cependant il constate que les scripts sont parfois si intrinsèquement liés à l'environnement dans lequel ils se déploient, qu'ils souffrent par conséquent d'un grand manque d'interopérabilité. Afin d'aborder une piste de solution à cette problématique Kobbe identifie les composants et les mécanismes génériques des scripts CSCL.

Composant d'un script CSCL selon Kobbe :

1. *Ressources* : Les objets physiques ou virtuels (ex : production verbale) alloué à chaque apprenant selon le script. Ces objets apportent de l'information ou offre une fonctionnalité. Ils peuvent être prédefinis et intégrés dans l'activité ou alors considérablement modifiés ou produits au cours du processus de l'activité.
2. *Participants* : Acteurs du script. En CSCL il est important de préciser leurs nombres (au total, dans chaque groupe etc.) et leur(s) rôle(s).
3. *Groupes*: Défini en général une structure hiérarchique, un groupe pouvant se composer d'un ou plusieurs groupes de tailles inférieures. Un acteur peut appartenir à plusieurs groupes avec différents rôles.

4. *Rôles* : De manière opérationnelle les rôles permettent d'attribuer les tâches à des individus en fonction de leurs rôles. Cependant les rôles sont aussi un moyen d'associer légitimité, privilège, obligations et attentes à un acteur. Kobbe donne l'exemple du rôle de critique ayant la légitimité de critiquer le travail d'un autre apprenant, le privilège de parler avec franchise, l'obligation de justifier sa critique et dont on attend de lui la mise en exergue de certains points pouvant donner lieu à des améliorations.
5. *Activités* : Un workflow scripté d'actions. Une activité peut-être décomposée en un ensemble de sous-activité selon une structure hiérarchique.

Mécanisme d'un script CSCL :

1. *Formation de groupes* : Certains groupes existent par définition et sont inchangeable (ex : Genre) d'autre peuvent être créé sur la base d'une procédure établie (diviser un groupe en 4, diviser un groupe en groupe de 4). Cette répartition peut se baser sur des critères autres que le nombre : équilibration des niveaux dans chaque groupe, répartition des compétences etc.
2. *Séquencement* : Le cœur du script, il forme la structure temporelle des interactions, le déroulement des activités. Il se base sur un modèle théorique d'apprentissage et d'enseignement. Il n'est pas forcément linéaire mais peut inclure bouclage et branchement.
3. *Distribution des tâches*: distribution des tâches entre les participants en fonction des rôles et du séquencement.

Si les langages graphiques de modélisation (voir point suivant) sont souvent basés ou interagissent avec le standard IMS-LD, certains s'intéressent aux scripts CSCL. Ainsi le projet COLLAGE (Hernández-Leo et al., 2006) propose un éditeur graphique basé sur des scripts CSCL en tant que « patron de conception » encapsulant les bonnes pratiques.

3. LANGAGE DE MODÉLISATION PÉDAGOGIQUE

3.1 Définition d'un LMP.

D'une manière générale, un langage de modélisation pédagogique (LMP ou EML - Educational modelling language en anglais) est un formalisme indépendant de toute infrastructure permettant de décrire, concevoir et éventuellement implémenter des scénarios pédagogiques. De manière plus précise le CEN/ISSS WS/LT Learning Technologies Workshop proposa de définir un LMP comme : « *un modèle d'information et d'agrégation sémantique, décrivant les contenus et les processus engagés dans une unité d'apprentissage selon une perspective pédagogique et dans le but d'assurer la réutilisabilité et l'interopérabilité.* » (Rawlings, Van Rosmalen, Koper, Rodriguez-Artacho & Lefrere, 2002, p.8). Les langages de modélisation pédagogique permettent de penser une situation de formation en s'intéressant au processus d'apprentissage de manière plus globale. En se focalisant sur l'activité, ce sont moins les ressources que la manière dont elles sont utilisées qui intéresse le concepteur pédagogique. Les LMP ont pour principaux objectifs :

- Permettre la modélisation d'un éventail de situation pédagogique aussi large que possible.
- De permettre la description, le partage, la discussion de scénario pédagogique entre concepteur, sur la base d'un langage commun.
- De faire le lien entre la dimension conceptuelle d'un scénario et son implémentation technique. Notamment en permettant ou facilitant la « projection » plus ou moins automatisée du scénario vers l'environnement informatique de formation.
- D'être un support cognitif pour le concepteur abordant la complexité d'un processus. (représentation graphique du processus, vu d'ensemble)

Selon Martel, Lejeune, Ferraris & Vignollet (2007, p.513) un LMP devrait avoir les 5 propriétés citées ci-dessous.

- P1 Posséder un niveau d'abstraction élevé permettant la spécification des activités indépendamment des contenus et des services propres aux ENT
- P2 : Permettre d'exprimer la diversité des rôles joués par les enseignants et les élèves au cours des différentes phases de l'activité pédagogique. L'organisation, l'observation, l'apprentissage ou l'évaluation supposent des rôles différents adoptés par les participants au cours de l'activité. Un élève doit pouvoir devenir tuteur d'un autre élève, un enseignant être tantôt le médiateur, tantôt un correcteur ou un simple observateur passif.
- P3 : Situer les échanges dans les différents contextes où ces échanges ont lieu. Comme dans les interactions de la vie quotidienne, les échanges sont caractérisés par le cadre dans lesquels ils se déroulent, par leur finalité ainsi que par le statut des utilisateurs qui y participent.
- P4 : Permettre d'exprimer les différentes relations possibles entre des activités résultant de scénarios distincts.
- P5 : Être métaphoriquement aussi neutre que possible pour permettre l'expression de multiples activités d'une manière générique.

Si IMS-LD est aujourd'hui une référence incontournable en modélisation pédagogique du fait de son statut de standard, les langages de modélisation pédagogique ont donné lieu à de nombreux travaux. Dans la section suivante nous présentons un panorama des différents LMP.

3.2 Panorama des recherches en LMP.

La recherche sur les langages de modélisation a donné lieu à de nombreux travaux du point de vue de :

- la théorie des langages (Cook, 2002; Ferruci, Tortora & Vitello, 2002) et langages visuels (Ferri, 2008 ; Marriott & Meyer, 1998).
- l'ingénierie pédagogique (Koper, 2002 ; Paquette, 2004 ; Botturi & al., 2006;) et ingénierie des EIAH (Baker, 2006 ; Choquet, 2007)
- et surtout en sciences informatiques où les langages visuels sont utilisé pour la modélisation de systèmes ou programmation d'applications (voir bibliographie de recherches sur les langages visuels effectuée par Burnett, 2009).

À la suite d'OUNL-EML devenu IMS-LD des langages de modélisation pédagogique ont été développés. Certains de ces langages sont en lien direct avec IMS-LD d'autre non. Certains se conçoivent comme des « ponts » vers l'implémentation technique tandis que d'autres ont pour unique vocation d'être des représentations descriptive en support du concepteur. Afin de pouvoir situer le travail effectué dans ce mémoire par rapport aux recherches dans le domaine de la modélisation pédagogiques nous présenterons succinctement ci-après différents LMP.

Ce panorama sera composé de *CompendiumLD* (Brasher & al., 2008), *PALO* (Rodríguez-Artacho & Verdejo Maíllo, 2004), *MOT* (Paquette et al., 2004, 2005), *LAMS* (learning activity management system - Dalziel, 2003) *E²ML* (Educational Environment Modeling Language - Botturi, 2008), *coUML* (cooperative Unified Modeling Language - Derntl et Motschnig-Pitrik, 2008), *poEML* (perspective oriented Educational Modelling Language - Caeiro-Rodriguez, 2008), *ASK-LDT* (Sampson, Karampiperis & Zervas, 2005), *LDL* (Learning Design Language - Martel & al., 2006), *COLLAGE* (Hernández-Leo et al., 2006), et *CPM* (Computer Problem-based Meta-model - Nodenot, Lafourcade et Le Pallec, 2008).

CompendiumLD :

C'est un outil de conception pour la scénarisation pédagogique utilisant un langage semi-formel assez simple. À l'aide de ce langage il est possible d'exprimer des scénarios pédagogiques à différents niveaux de granularité (ex : niveau formation, niveau cours, niveau activité etc.).

CompendiumLD a pour objectif d'aider les concepteurs à élaborer des environnements techno-pédagogiques en formalisant graphiquement la mise en relation des activités, des acteurs, des services et des objets. Un scénario réalisé avec CompendiumLD se présente sous la forme d'un diagramme « type carte conceptuelle » allant du haut vers le bas (ordre temporelle). Chaque carte est composée de nœuds typés (tâches, objectifs, outils, acteurs, ressources, etc.) reliés entre eux par des liens. Compendium offre de plus la possibilité d'inclure des cartes dans des cartes via un noeud de connexion spécifique et d'éditer chacun des nœuds avec des descriptions longues non visibles directement sur la carte.

Figure 10: CompendiumLD illustration.

CompendiumLD est purement descriptif. Il ne permet pas de créer du code exécutable, ni d'exporter au format IMS-LD. Les cartes peuvent être exportées en XML, archive .zip, jpeg et en html pour lecture dynamique sur le web. CompendiumLD est encore mise à jour et développé depuis sa création en 2003 dans le cadre Open University Learning Design Initiative.

E²ML

E²ML est un langage de modélisation développé spécifiquement pour la conception des environnements éducatifs. Il a surtout été conçu pour des spécialistes de l'éducation ayant une bonne connaissance des modèles d'Instructional Design afin de documenter efficacement les dispositifs dans les projets E-learning importants. Le modèle E²ML est compatible avec la spécification IMS Learning Design et peut être intégré à des normes de métadonnées d'objets d'apprentissage. En revanche il ne produit aucun code exécutable.

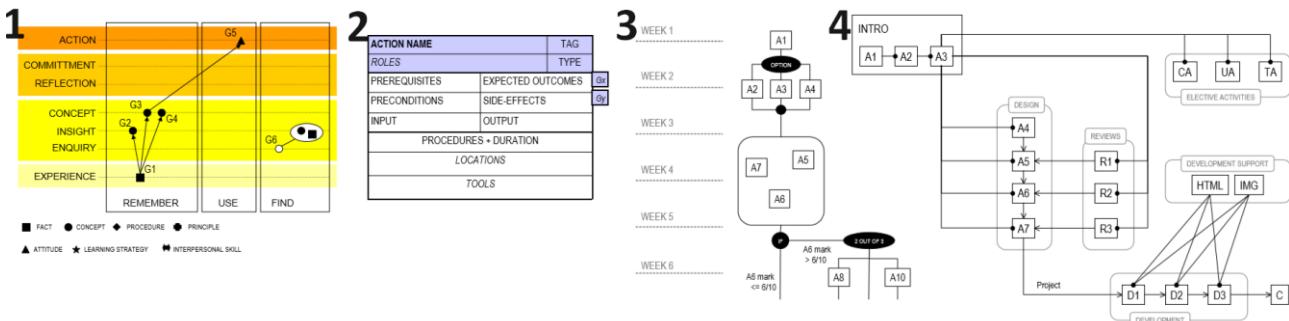


Figure 11: Les 4 types de diagrammes E²ML (Botturri, 2003)

Le langage est composé d'un ensemble de diagrammes interdépendants permettant la cartographie des objectifs d'apprentissages puis la représentation visuelle des activités d'apprentissage. Nous retrouvons donc :

1. La définition des objectifs d'apprentissage composée d'une déclaration verbale et d'une cartographie visuelle (le modèle QUAIL). Le modèle QUAIL a été conçu en se basant sur des modèles d'ID (Taxonomie de Bloom, classification de Gagné, matrice de Merrill).
2. Le diagramme d'action se présente sous la forme d'un tableau normalisé représentant la plus petite unité du dispositif et pouvant être relié par des identifiants (côté droit du tableau) aux objectifs du

modèle QUAIL précédent. Une action est un petit ensemble de tâches effectuées en vue de l'atteinte d'objectifs d'apprentissage.

3. Le diagramme de flux permet de séquencer temporellement les actions et de spécifier différents chemins empruntés par les apprenants.
4. Le diagramme de dépendances. Il permet de relier les blocs d'action définie à l'étape 2. Cela permet d'identifier (1) les prérequis par des liens terminés d'un point, (2) les produits d'une action nécessaire à une autre action par une flèche et (3) d'agréger les actions en unités de formation par des blocs d'encadrement.

LAMS

L'outil **LAMS** (Learning Design Management System) est un outil graphique qui permet de créer et d'exécuter des scénarios conformes à IMS LD niveau A. LAMS offre ainsi un environnement en ligne composé d'un outil

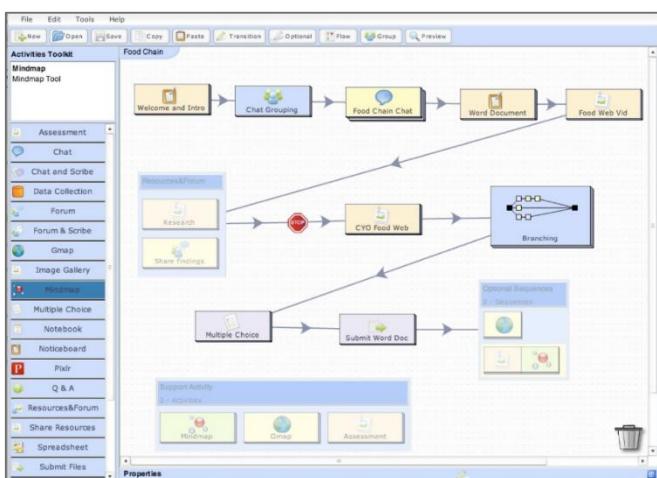


Figure 12: Environnement autour de LAMS dans un navigateur.

auteur et d'un moteur d'exécution permettant de présenter et gérer des activités dans un ordre précis à l'aide d'une interface "drag and drop". La notation visuelle utilisé par LAMS est très basique et se compose simplement de « boîtes » rectangulaires représentant des tâches à effectuer par l'apprenant généralement en relation avec un outil informatique (chat, wiki, forum, dépôt de devoir, QCM etc.). Ces boîtes sont ensuite connectées par des flèches représentant le déroulement (uniquement linéaire) de la séquence d'apprentissage et des verrous (« panneaux stop ») permettent d'ajouter des conditions à satisfaire avant de passer à la suite. De

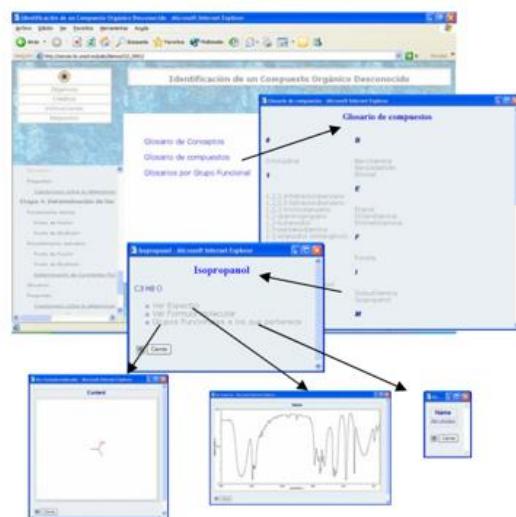
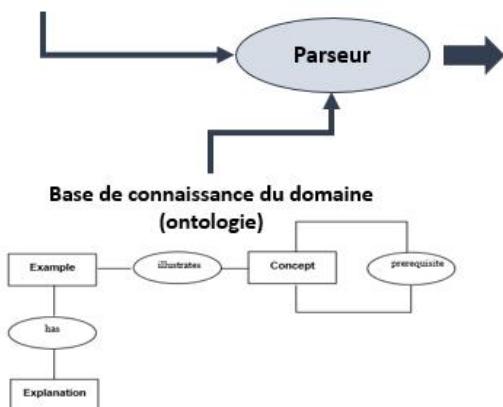
ce fait, considérer LAMS comme un LMP peut être discutable. Toutefois LAMS permet réellement l'échange de séquences pédagogiques entre concepteurs. De plus LAMS représente la seule réelle technologie permettant de manière simple, stable et gratuite d'exécuter des activités pédagogiques selon un certain séquencement. De plus LAMS offre la possibilité d'importer dans certains LMS (ex : Moodle, Blackboard) des activités LAMS dont le formalisme est proche du standard IMS-LD. Cependant une telle intégration reste encore superficielle, les activités LAMS utilisant leurs propres services et n'interagissant pas avec les services du LMS. LAMS a été créé en 2003 et continue d'être développé. La dernière mise à jour est la version 2.4 sortie en avril 2012.

PALO

PALO est un langage de modélisation pédagogique non visuel (comme IMS-LD) permettant de décrire des contenus pédagogiques et des environnements d'apprentissage avec un haut niveau d'abstraction. L'approche de PALO consiste à décrire des cours structurés en modules selon une perspective ouvertement cognitiviste car basé sur des ontologies de connaissance. Chaque module inclut une déclaration de sa structure, des activités que les acteurs doivent réaliser, ainsi que le séquencement des activités et des contenus (Rodríguez-Artacho & Verdejo Maíllo, 2004, p.10). Cette déclaration s'effectue via un fichier SGML contenant la description de l'environnement d'apprentissage en accord avec un modèle de grammaire (DTD). La description PALO d'un contenu pédagogique (fichier *.palo) peut être déployée en une variété de scénarios pédagogiques après compilation. L'utilisation conjointe d'un parser, d'un compilateur, d'ontologie du domaine et d'une DTD permet de générer un environnement d'apprentissage utilisable en ligne ou hors -ligne (via un applet Java).

Modèle PALO (DTD)

Fichier PALO



Contrairement à IMS-LD, PALO n'est donc pas un langage neutre concernant l'approche pédagogique. La déclaration de contenu se base sur un modèle d'enseignement spécifique (« instructional template ») qui se traduit dans une DTD (définition de type de document). Cette approche par modèle est une aide appréciable pour la création de contenus respectant une pédagogie cohérente. Le problème principal est que PALO ne propose que très peu de modèles mais il permet et encourage leur création en facilitant l'extension du langage. Si PALO semble avoir été utilisé activement dans le cadre de cours donnés par l'université de formation à distance de MADRID jusqu'en 2004, il est difficile de trouver des informations sur le sujet depuis. Il est possible que son développement ait été abandonné.

CoUML

CoUML est un langage de modélisation pédagogique pouvant être utilisé pour modéliser l'apprentissage coopératif dans les dispositifs instrumentés. CoUML signifie « Coopérative UML », indiquant que son système de notation est essentiellement une extension d'UML utilisé pour modéliser des activités coopératives. Étant basé sur UML, il offre un système de notation formelle et théoriquement neutre (en pratique il reste cependant « imprégné » du paradigme de programmation orienté-objet issu des pratiques du génie logiciel). De manière assez semblable à E²ML, le langage coUML se présente sous la forme de différents diagrammes.

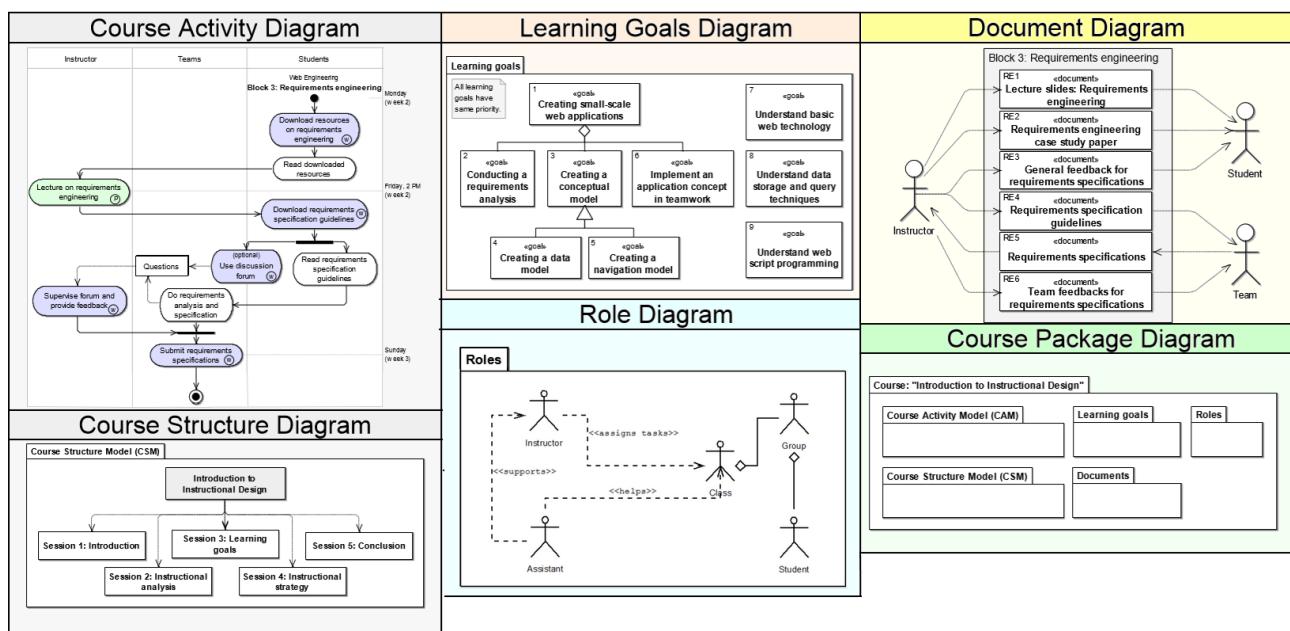


Figure 13: Illustration des diagrammes de coUML (adapté depuis Figl & al. 2010)

CoUML est particulièrement indiqué pour les formations hybrides avec déroulement d'activités collaboratives complexes. Cependant le langage ne fournit aucun code exécutable (même si la base UML laisse la porte ouverte à un développement en vue d'une implémentation) et aucun export vers le standard IMS-LD. La communauté autour de ce langage est réduite et aucun outil spécifique n'existe pour la modélisation coUML. Il faut utiliser un outil de modélisation pour UML.

PoEML :

PoEML est un langage de modélisation visuel orienté perspectives (Perspective-oriented Educational Modeling Language). L'idée sous-jacente est qu'au lieu de résoudre la modélisation d'un processus d'apprentissage dans son ensemble, ce dernier peut être décomposé en parties indépendantes (« les perspectives »). La décomposition est un principe connu d'ingénierie permettant d'aborder les problèmes complexes en les segmentant puis en focalisant l'attention sur chacune des parties. Du point de vue de la modélisation cette décomposition permet de maintenir un niveau d'expressivité élevé d'un langage tout en préservant sa simplicité, flexibilité, adaptabilité et sa réutilisabilité. Il est par exemple ainsi possible d'effectuer des modifications à un endroit sans briser l'ensemble. PoEML est un langage formel conçu en vue d'une implémentation, l'idée de pouvoir effectuer des changements « à la volée » d'une partie du système sans affecter l'ensemble est donc particulièrement attractive. L'entièreté du langage est définie via UML.

En se basant notamment sur les concepts de la théorie de l'activité les auteurs identifient 13 perspectives composant un scénario pédagogique exécutable (Voir modèle UML de l'annexe 3). Chaque perspective est un « package » possédant un petit jeu de symboles graphiques spécifiques. Les symboles sont connectés entre eux (association, agrégation, spécialisation et dépendance) via des flèches identiques à celles utilisées dans les diagrammes UML. Enfin ces perspectives sont exprimées dans poEML selon 4 niveaux d'abstraction (les « aspects »).

1. Le niveau le plus élémentaire porte sur les données manipulées c'est-à-dire des éléments de données simples (booléens, entiers, etc.), structurés (fichiers, pages Web, etc.), relatives aux participants (données de profil, productions etc.).
2. Le second niveau porte sur la gestion de l'exécution du modèle en fonction de certaines conditions ayant trait aux valeurs des données.
3. Le troisième niveau porte sur le contrôle de l'exécution du scénario en fonction des décisions de participants selon le scénario prévu durant la modélisation.
4. 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.

LDL

Learning Design est un langage formel de modélisation pédagogique qui a été conçu pour modéliser des activités collaboratives. Le langage est basé sur sept concepts de base. [1] *Le Scénario* (séquencement), [2] *la Structure* (définit les composants), [3] *les Interactions* (échanges verbale ou matériel entre participants), [4]

l'enceinte (lieu, ou service informatique où se situe l'action), [5] *les Rôles*, [6] *les observables* (tableau de bord, trace d'activité) et [7] *les Positions* (Points de vue des participants sur une entité du cours). La démarche de conception d'un scénario avec LDL repose sur :

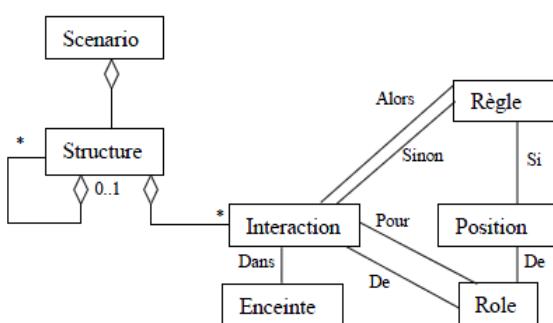


Figure 14: Diagramme UML simplifié de la structure de LDL

1. l'identification et la construction de sa structure interactionnelle, c'est à dire de la manière dont les échanges vont s'organiser et se dérouler ;
2. la définition des différents rôles qui prendront part à l'activité ;

3. la définition des enceintes qui sont les lieux dans lesquelles va se dérouler l'activité ;
4. la définition des règles auxquelles les participants vont se conformer ;
5. la définition des positions des participants, c'est à dire des différents points de vue qu'ils auront à exprimer au cours de l'activité.

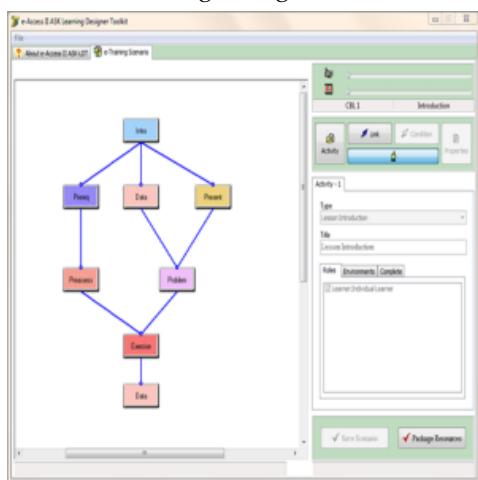
Les scénarios réalisés semblent pouvoir être exécutés via une infrastructure d'exécution (LDI - Learning Design Infrastructure). Cette dernière fonctionne sur la base d'un serveur libre programmé en python d'application web orienté objet (Zope) et d'un environnement de gestion et de contenus (documents, e-mail, messages textuels, etc.) basé sur le module complémentaire Plone. Un moteur d'exécution (LD Engine) exécute des requêtes sur le serveur pour récupérer observables, positions, interactions et les comparer aux règles, rôles etc.

LDL et LDI semblent encore être à un stade expérimental.

ASK-LDT

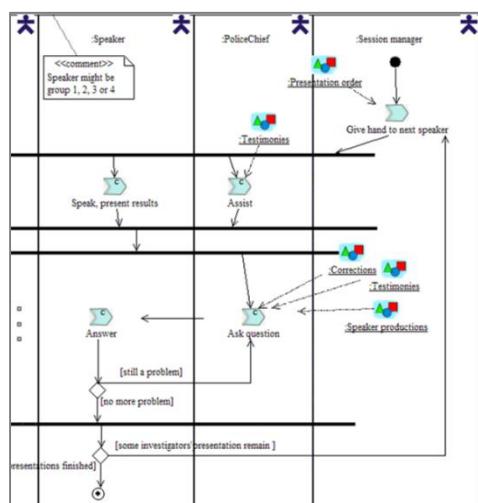
ASK Learning Designer Toolkit est un outil graphique permettant de représenter des interconnexions

d'activités suivant la spécification IMD-LD. L'outil ASK-LDT propose au concepteur de relier blocs de construction mettant en relation les actions respectives des apprenants et des enseignants. Selon (Nodenot & Al., 2008) il s'agit ainsi de contextualiser dans le cadre de scénario des blocs « exploration/documentation », « experimentation/reactivity », « debate/ animation », « self-reflection/ co-reflection » etc. Les scénarios créés avec ASK-LDT peuvent être exportés sous forme de module « content pack » conforme à la norme IMS CP et IMS LD. Il est difficile de trouver de l'information concernant la poursuite du projet ASK-LDT et nous ne savons pas si le développement du logiciel est encore actif.



CPM

A l'instar de CoUML, CPM (Computer Problem base Metamodel) est un langage visuel construit comme une spécialisation du langage UML. Il en reprend les mécanismes afin de poursuivre la description de diagrammes



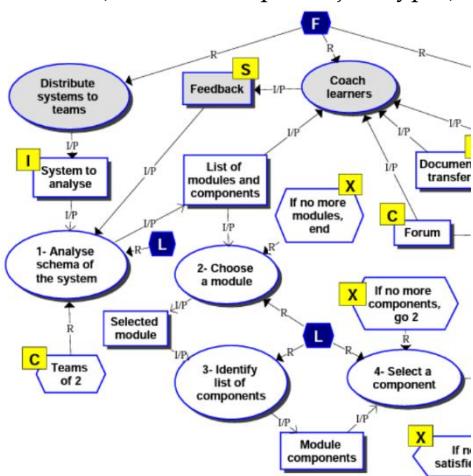
descriptifs ou exploratoires, vers le développement d'une solution opérationnelle. Comme son nom l'indique CPM se focalise sur des scénarios socioconstructivistes de type « apprentissage par problèmes ». Il s'adresse à des ingénieurs-logiciels spécialisés dans le domaine de l'ingénierie éducative. Il permet la conception de diagrammes visuels décrivant des situations problèmes. Ces diagrammes peuvent ensuite être convertis vers le niveau A du langage IMS-LD. Selon les auteurs, les éléments proposés 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. Les scénarios réalisés avec CPM se basent sur un ensemble de diagrammes interdépendants.

Typiquement un processus de conception avec CPM démarre «par la production de diagrammes de use cases (type de diagramme UML). Chaque diagramme est ensuite raffiné par d'autres diagrammes de use cases puis par un, voire plusieurs diagrammes d'activité (Autre représentation UML) 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 séquencement de ces tâches » (Nodenot, Laforcade et Le Pallec, 2007).

MOT et MISA

MOT (Modélisation par Objets Typés) est une notation graphique semi-formelle (dispose d'une traduction en XML) permettant la mise en œuvre de la Méthode d'ingénierie MISA.



MISA est une méthode d'ingénierie très complète mais complexe et très industrielle de soutien à la conception des dispositifs d'apprentissage. Elle s'adresse à un public d'expert.

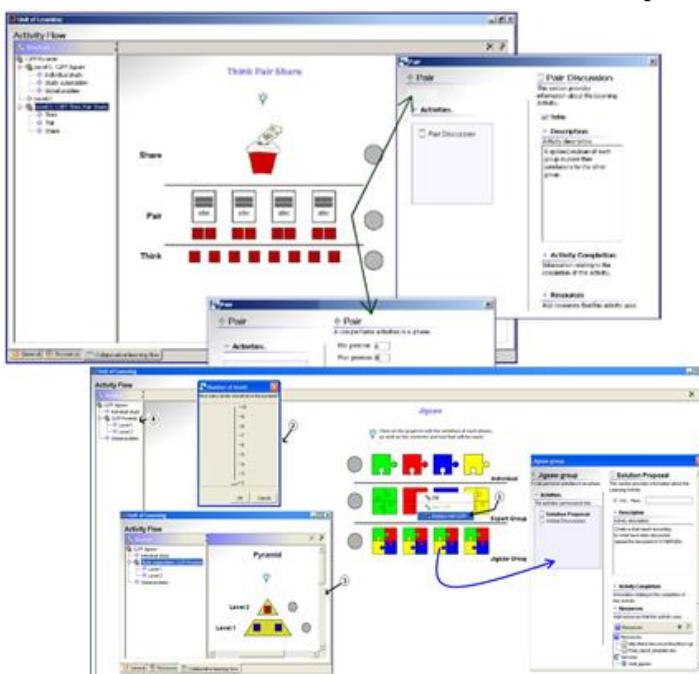
Le modèle pédagogique de MISA représente un scénario pédagogique comme un réseau d'événements d'apprentissage. MOT permet la représentation de ce réseau composé d'activités (Ovale), de ressources (rectangle) et de règles (hexagone). Des lettres permettent de spécifier les liens (ex : P = « connexion de type prérequis », R = « connexion de type règle ») ou tout autre forme graphique (ex : T = « Tool », C = « Instrument », L = « Learner » F = « Facilitator »). Le concept de réseau est le cœur de l'approche

MOT qui très influencé par la conception d'ontologie de connaissance.

L'éditeur graphique MOT+LD permet de construire des scénarios pédagogiques avec le langage MOT conforme au manifeste XML de IMS-LD niveau A.

COLLAGE

Collage est un éditeur graphique de scénarios d'apprentissage collaboratif, basé sur l'exploitation des patrons de scénario issu des CSCL (Alexander & al., 1977). Le principe des patrons est « d'encapsuler » les bonnes pratiques de ses contributeurs dans des « patrons pédagogiques » permettant à un enseignant de modéliser des activités collaboratives à haut niveau en commençant par des modèles existants. Ainsi le modèle JIGSAW scripte le déroulement d'une activité d'apprentissage collaboratif dans laquelle différents groupes sont face à l'étude de nombreuses informations pour la résolution d'un problème commun complexe. Le scénario est ensuite exportable en un package conforme au standard d'IMS-LD niveau A.



3.3 Synthèse.

Ce panorama des travaux sur les LMP dénote un certain intérêt pour le sujet mais révèle également la diversité des approches et les difficultés restant à surmonter.

De manière synthétique les LMP se distinguent :

- *Par leur niveau de formalisation* : de totalement informelle à des langages définis par une véritable sémantique et syntaxe (souvent basé sur XML).
- *Par leurs perspectives* : Certains langages proposent d'aborder une situation pédagogique selon un seul angle d'approche, un seul aspect, d'autres permettent la modélisation d'une même situation sous différentes perspectives (ex : interactions des apprenants et structure des contenus). Ces différentes perspectives s'accompagnent généralement de diagrammes aux graphismes et mécaniques différentes. Les différents diagrammes peuvent être interdépendant (lié) ou non.
- *Par leurs niveaux de granularité* : Certains sont des langages « à plat », d'autres permettent de décomposer les modélisations en différents niveaux de granularité supérieur ou inférieure.
- *Par leur orientation pédagogique* : Certains langages se veulent neutre pédagogiquement, d'autre vise la modélisation liée à une pédagogie particulière. (Nous parlons ici d'une « volonté de neutralité », nous pensons personnellement qu'un langage n'est jamais neutre quant à l'approche).
- *Par leur format* : Les langages peuvent être graphiques ou textuels (voir les deux).
- *Par leur finalité* : La finalité donne un sens au langage. Un langage ne se conçoit pas de la même manière s'il est destiné à la documentation, à la communication, à la créativité, à l'implémentation d'un dispositif etc.
- *Par leur relation aux standards* : Un langage peut être un standard (IMS-LD), être basé sur un standard (UML, XML, IMS-LD), être convertible vers un standard (ex : IMS-LD niveau A) ou enfin n'entretenir aucun lien.

Les travaux de modélisation pédagogiques développés jusqu'ici offrent des perspectives intéressantes mais nous constatons cependant qu'ils ne rencontrent pas encore le succès auquel ils aspirent. La spécification IMS-LD est assez complexe et les ingénieurs pédagogiques passent beaucoup de temps et d'efforts juste pour être en mesure de définir un scénario pédagogique (Morales & al. 2008). Les autres langages de modélisation (notamment graphique) spécialement développés pour l'éducation restent souvent au stade expérimental et souffrent d'un manque d'outils, de documentations et d'une communauté de soutien pour réellement être opérationnels. De plus lorsque des langages graphiques de conception permettent un export vers IMS-LD ce dernier est souvent partiel et ne dépasse en aucun cas le niveau A de la spécification.

4. INGÉNIEUR PÉDAGOGIQUE ?

4.1 Artisanat et ingénierie.

Si l'informatique offre des possibilités d'automatisation, de duplication, de diffusion et de réutilisation quasiment illimitées elle comporte souvent un coût en ressource (temporelle ou financière) important. Dès lors, apparaît la volonté d'amortir ce coût sur le nombre et la durée d'usage ou sur la réutilisation potentiel du dispositif en favorisant ses aspects génériques et interopérable. Cependant ce processus demande de s'extraire d'une posture d'artisan *bricoleur* pour se rapprocher de celle de l'*ingénieur* (Lévi-Strauss, 1962). Pourtant, la posture de l'enseignant a historiquement toujours été plus proche du *bricoleur*. En effet, l'enseignant agit au cas par cas, adaptant ce qu'il a « sous la main » aux situations diverses d'apprentissage. Typiquement, il analyse une situation, identifie un problème, puis se tourne vers son « atelier » où il puise dans l'ensemble des ressources, outils et méthodes pédagogiques dont il dispose ce qu'il pense avoir besoin afin de créer son enseignement.

L'arrivée des TICE n'a pas fondamentalement bouleversé ce processus. Comme le constate Pierre André Caron (2007) :

« Les enseignants ne disposant pas d'ingénieur pédagogique, utilisent des applications conçues et administrées de façon indépendante de la pédagogie qu'ils désirent mettre en place. Cette démarche qui pour un enseignant consiste à faire "avec les moyens du bord" (avec les applications Web disponibles) s'apparente plus à une démarche de bricolage qu'à une démarche ingénieriste. [...] Choisir une application Web 2.0, parce qu'elle est disponible et parce qu'elle se rapproche des fonctionnalités qu'il désire utiliser constitue pour l'enseignant une activité que nous pouvons décrire par du bricolage pédagogique. » (p.138)

Il existe donc une certaine tension entre une posture enseignante habituée à fonctionner au cas par cas - créant des dispositifs à l'investissement initial faible et à usage unique qu'ils tentent d'optimiser pour le contexte auquel ils sont confrontés – et une certaine volonté de généréricité rationalisant les investissements plus importants.

De manière schématique, une approche artisanale en e-learning est souvent caractérisée par des outils auteurs, des progiciels d'édition et des services web 2.0, offrant aux enseignants/concepteurs une *boîte à outils* susceptible de répondre à diverse contextes de formation. Cette solution est proche de la posture traditionnelle de l'enseignant qui construit pour chaque situation une *œuvre unique* pour une situation spécifique aux besoins bien identifiés. Cette production est utilisée dans le cadre d'un seul cours et est réutilisé moyennant de faibles ajustements si ce même cours se répète dans le temps. Cette approche aboutit généralement à de petit module e-learning (dont la qualité et l'efficacité pédagogique est très dépendante des compétences du concepteur) généralement diffusé via un LMS.

Une approche d'ingénieur quant à elle s'inspire de méthodes plus industrielles. Cette approche tente de s'extraire des spécificités de la situation d'apprentissage pour abstraire des modèles génériques ou automatiser certaines tâche de développement. Une telle démarche se justifie généralement par la complexité ou la taille importante de certains projets de formation. D'une part nous observons une demande pour des EIAH innovant dans des domaines de pointe tels que l'aéronautique, la médecine etc. dont la complexité ne peut être abordée via des méthodes artisanales. D'autre part, l'évolution des pratiques de formations actuelles soumet la conception des dispositifs médiatisés à des contraintes logistiques, techniques, pédagogiques immenses, en rupture avec les unités traditionnelles de temps, de lieux et d'actions (Carre, Cours d'Ingénierie Pédagogique, 2008). Le développement actuel des Massives Online Open Courses (MOOC) stigmatise à lui seul les contraintes opérationnelles auxquelles se confrontent les concepteurs aujourd'hui. Former un nombre croissant d'individus tout en permettant une individualisation des parcours. Maintenir l'organisation et la cohérence d'une formation tout en assurant une flexibilité physique et temporelle pour des apprenants en activité professionnelle. Rendre l'environnement flexible, personnalisable et ergonomique tout en restant évolutif et facile à maintenir. Développer de réelles compétences malgré une restriction des possibilités d'accompagnement humain... La liste pourrait être longue.

4.2 Dispositifs instrumentés et transformations des acteurs.

« L'introduction des TICE dans l'ingénierie de formation est à prendre comme une donnée indépendante de tout jugement de valeur. Nous ne voulons pas évaluer ici un événement historique des Sciences et Techniques, ni aucune de ses conséquences sociales et idéologiques. Cela représente un fait dont nous prenons acte: les TICE sont la toile de fond de la nouvelle ingénierie de formation, et ne sont plus simplement conjoncturelles. Elles sont une des caractéristiques structurelles de la situation de formation présente et à venir. »

[Delacour, 2008, p.126]

Les situations pédagogiques instrumentées par les TICE sont aujourd’hui incontournables, que ce soit dans le contexte des institutions ou dans celui des entreprises. Dès lors l’apprentissage se noue au cœur d’un dispositif technopédagogique intégrant de manière croissante les dimensions ligne/hors-ligne, distance/présence, individuel/collaboratif, formel/informel, théorie/pratique afin d’enrichir les modes de formation « traditionnels » et ce avec un rapport qualité/prix raisonnable (Singh, 2003). Si un dispositif peut simplement être défini comme « un ensemble de moyens humains et matériels agencés en vue de faciliter un processus d’apprentissage » (Blandin, 2002), rappelons que ce dernier repose sur des formes complexes de médiatisation et de médiation (Charlier, Deschryver, Peraya, 2006) et qu’il s’incarne avant tout dans des acteurs agissant (Linard, 1998). Un dispositif est pensé, conçu, déployé puis fonctionne selon une dynamique propre aux acteurs en interaction à l’intérieur de celui-ci. Dans un dispositif instrumenté, l’enseignant est avant tout, un concepteur.

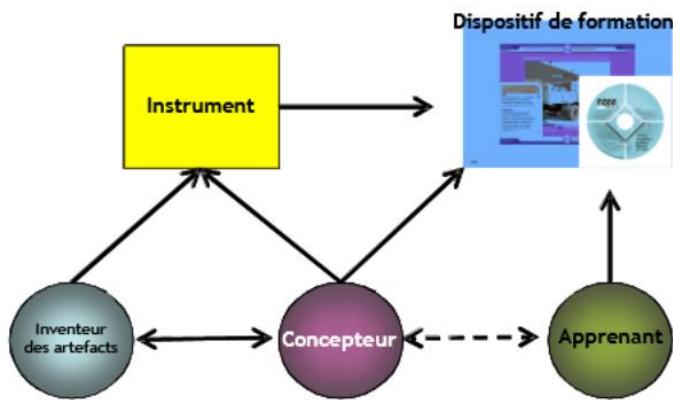


Figure 15: Schéma Enseignant concepteur selon Delacour (2008)

Comme nous l’avons abordé au point précédent, l’accroissement de la taille, de la complexité et de la technicité de certains dispositifs instrumentés peut nécessiter de s’orienter vers des méthodes normalisées d’ingénierie. À tel point qu’il est courant de rencontrer dans la littérature comme sur le terrain, la présence d’un acteur emblématique de la conception de formation : l’ingénieur pédagogique. Si son rôle est encore inconnu ou flou, cet acteur est de plus en plus à l’œuvre au sein des dispositifs de formation. Il est un acteur « charnière » qui possède « *des compétences pédagogiques et techniques et réalise le lien entre les deux mondes* » (Vantroys & Peter, 2005). Nous pouvons distinguer deux raisons principales à la montée en puissance de cet acteur au sein de la « vague e-learning » :

- Premièrement, la nécessité de recentrer l’approche de conception des dispositifs instrumentés autour de la pédagogie. Comme le constate Delacour (2008) le formateur a parfois purement et simplement disparu des dispositifs e-learning. Selon lui, « *de nombreux programmes e-learning ont été réalisés entièrement par des équipes composées de spécialistes de la communication « corporate », d’informaticiens, ou de vidéastes, sans programmation pédagogique et sans consultation d’un formateur ni d’aucun pédagogue.* » (p.126).
- Deuxièmement, la nécessité d’une compréhension de la pluridisciplinarité au sein des dispositifs e-learning. Il est bien souvent réducteur d’analyser les dispositifs selon une dichotomie enseignant/apprenant. Les dispositifs de formation peuvent invoquer une diversité considérable d’acteurs : experts contenus, chefs de projet, formateurs, équipes de support, programmeurs informatique, développeurs multimédia, designers web, tuteurs, « community managers », administrateurs réseaux etc. L’ingénieur pédagogique se conçoit alors comme un acteur tissant des liens entre les différentes disciplines afin d’atteindre un objectif d’apprentissage respectant les intentions pédagogiques de départ. Il est à noter cependant, que lorsqu’il s’agit d’intégrer les technologies numériques dans une formation académique, différents rôles sont cumulés par un même acteur (l’enseignant-concepteur), les contraintes économiques ne permettant pas de l’accompagner par une équipe de concepteurs et de développeurs.

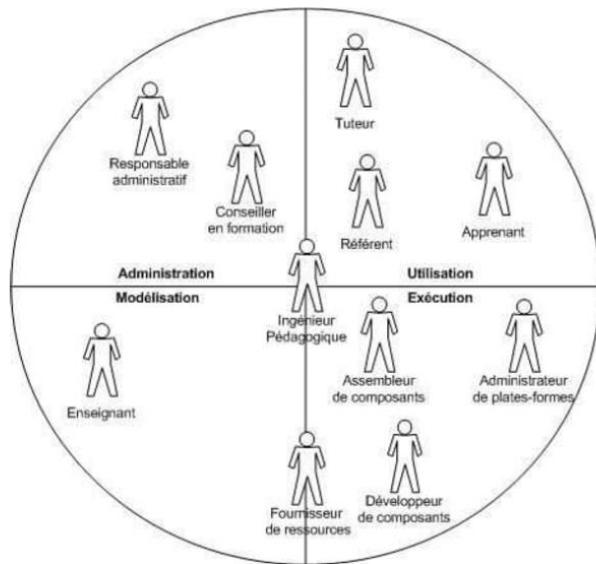


Figure 16: Acteurs et domaines fonctionnels d'un dispositif de formation à distance (Vantroys, 2005, p.6)

La conception des dispositifs instrumentés de formation revêt dès lors un fort enjeu communicationnel. C'est notamment une des perspectives que nous aborderons avec BPMN au cours des chapitres suivants. L'usage d'un langage graphique de modélisation connu à la fois des équipes IT, des équipes de directions d'entreprise et des ressources humaines nous paraît être intéressant pour qu'un ingénieur pédagogique communique efficacement.

5. INGÉNIERIE PÉDAGOGIQUE ET INGÉNIERIE DES EIAH

Selon Tchounikine (2004), « *Un EIAH est un environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant* » (p.1). Comme le souligne l'auteur, le terme renvoie ainsi à une multiplicité de types de systèmes. Un EIAH peut être composé d'un système informatique non initialement prévu pour l'apprentissage mais détourné pour un usage pédagogique. Le terme « EIAH » défini alors le «*système élargi* » comprenant l'artefact informatique et l'usage de celui-ci dans une visée d'apprentissage (p.2). Toutefois lorsqu'il s'agit de « *conception des EIAH* » la problématique est différente, on évoque alors la conception d'un artefact spécifique d'accompagnement de l'apprentissage « *embarquant en lui une intention didactique et/ou pédagogique* » (*ibid.*). Le développement d'EIAH relève donc d'une ingénierie à la croisée des sciences informatiques et des sciences de l'éducation. D'après Tchounikine (2002) l'ingénierie des EIAH peut être définie comme les « *travaux visant à définir des concepts, méthodes et techniques reproductibles et/ou réutilisables facilitant la mise en place (conception – réalisation – expérimentation – évaluation – diffusion) d'EIAH en permettant de dépasser le traitement ad hoc des problèmes* ». L'ingénierie des EIAH visent donc l'amélioration des façons de concevoir et d'implémenter des applications logicielles dans le cadre de formation à distance, hybride ou présentielle. Comme le rappelle De Moura Filho & Derycke (2007) concevoir n'est jamais une tâche humaine triviale:

« *Les concepteurs sont toujours en train de négocier avec le « nouveau, l'imprévu », ou comme le dit Schön (1987) avec une situation embrouillée et indéterminée. Parce ce que “la réalité humaine est trop complexe pour que nous puissions y agir sans la réduire” (Meirieu, 1987), les concepteurs, y compris ceux des EIAH, sont en permanence en train d'essayer de rendre leur tâche plus tangible, en réduisant la complexité, et tout en préservant ce qu'ils considèrent comme essentiel. Une façon de réduire la complexité de la tâche de conception est de contraindre l'espace problème, et pour arriver à cela, il est souvent fait appel à la puissance de l'abstraction : “One of the most important challenges of system design is dealing with complexity. We attack complexity with abstraction.” (Coplien, 1996). Une voie scientifique, et commune, de se frotter à l'abstraction, est d'utiliser des modèles.* » (p. 122)

La plupart des recherches récentes en EIAH tentent de définir des modèles conceptuels basés sur une vision pédagogique et les méthodes du génie logiciel. Le plus souvent, ces modèles conceptuels sont ensuite exprimés via un ensemble de spécifications plus ou moins formelles et standardisées accompagnées d'une interface de programmation, de gestion ou de service (une API). Le lien entre ingénierie des EIAH et ingénierie pédagogique est très étroit. Tout comme cette dernière, l'ingénierie des EIAH s'est longtemps focalisée sur des modèles documentaires (voir section 1.1) aboutissant au développement de plateforme de formation (type LMS – Learning Management System). Depuis les années 2000 et les recherches sur les LMP en ingénierie pédagogique, l'ingénierie des EIAH s'intéresse aux méthodes de conception d'environnement informatique centrées sur la modélisation de scénarios pédagogiques. Le schéma que nous proposons en *figure 5* peut être modifié pour représenter la relation triangulaire de la scénarisation sous le paradigme des dispositifs instrumentés et des méthodes d'ingénieries (figure 17).

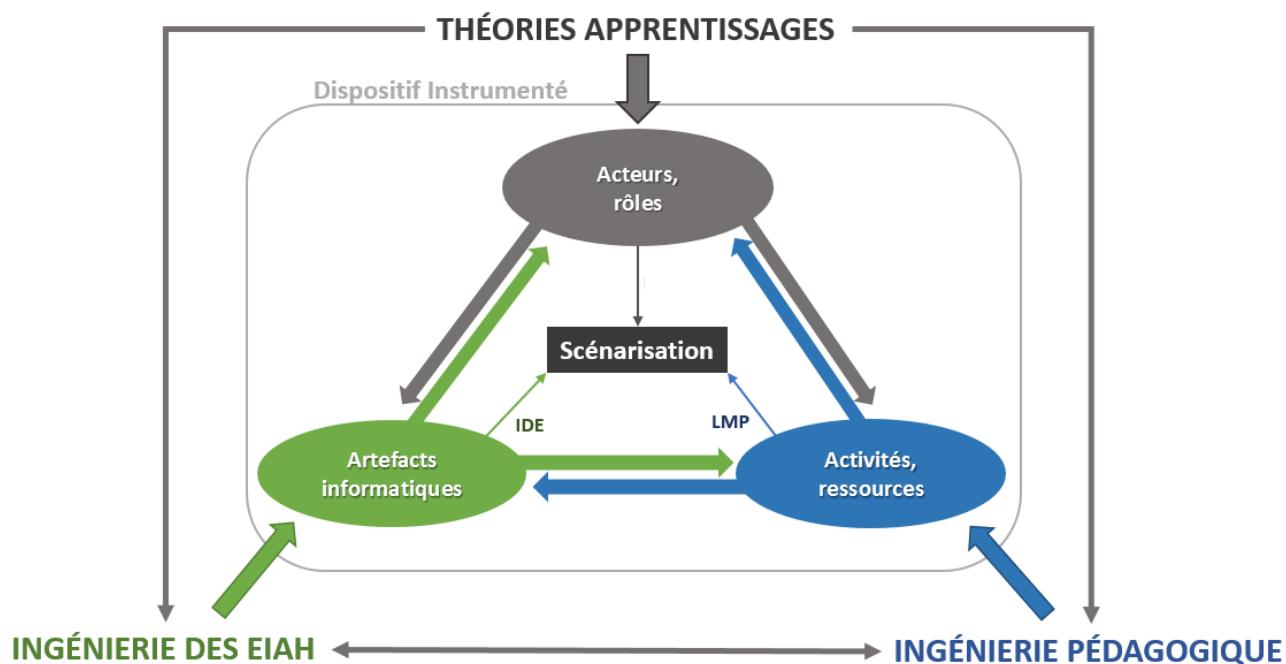


Figure 17: Schéma récapitulatif - la scénarisation sous le paradigme des dispositifs instrumentés

5.1 Ingénierie dirigée par les modèles.

Modéliser de manière formelle une situation d'apprentissage en vue de générer un EIAH fit rapidement écho au paradigme IDM (Ingénierie Dirigée par les Modèles) en plein développement au sein des sciences informatiques. Ce paradigme de conception nous intéresse particulièrement pour notre approche de modélisation avec le langage BPMN. Avec la généralisation des formalismes et langages informatiques orientées objet, l'activité de modélisation prend encore plus de poids, car elle peut servir directement la génération d'applications informatiques, tel que des EIAH dans notre cas. L'IDM, connue sous le terme MDE (Model-Driven Engineering) en anglais, représente une discipline où les modèles sont au cœur du processus de développement logiciel. En IDM les modèles cessent d'être simplement « contemplatifs » (documentation informatique) pour devenir « génératifs ». Le processus fondamental de cette approche réside donc dans la transformation de modèles. Selon Samba (2011) :

« Les modèles occupent une place de premier plan parmi les artefacts de développement des systèmes et doivent en contrepartie être suffisamment précis et riches afin de pouvoir être interprétés ou transformés par des machines. Le processus de développement des systèmes peut alors être vu comme un enchaînement de transformations de modèles, chaque transformation prenant un ou des modèles en entrée et produisant un ou des modèles en sortie, jusqu'à l'obtention d'artefacts exécutables. » (p.29)

Ces techniques favorisent particulièrement la réutilisation, les modifications et la maintenance puisque les interventions se font principalement au niveau des modèles et non plus au niveau du code (Favre, 2006).

Afin de pouvoir créer, manipuler, partager puis transformer les modèles jusqu'au code source, l'IDM a du se doter d'outil opérationnalisaant ce processus. On parle alors de métamodélisation. La métamodélisation a permis l'émergence des DSMLs (Domain Specific Modeling Language), que l'on peut traduire en français par « *langages de modélisation spécifiques à un domaine* ». Un DSML est défini par (*ibid.*):

- Une syntaxe abstraite qui décrit les constructions du langage et leurs relations ;
- Une syntaxe concrète qui décrit le formalisme permettant à l'utilisateur de manipuler les constructions du langage ;
- Une sémantique statique qui permet d'avoir un DSML clairement défini donc « outillable ».

L'intérêt pour l'IDM a été fortement amplifié dans les années 1990 lorsque l'organisme de standardisation Object Management Group (OMG) a rendu publique son initiative MDA (Model Driven Architecture). La MDA fait partie des approches de génie logiciel d'IDM. Il existe d'autres approches IDM tel que la programmation par aspects (AOP - aspect-oriented programming), les usines à logiciel (Software factory de Microsoft) ou encore le MIC (Model Integrated Computing). Cependant nous nous intéresserons particulièrement à la MDA car BPMN est développé par l'OMG dans le cadre de cette approche.

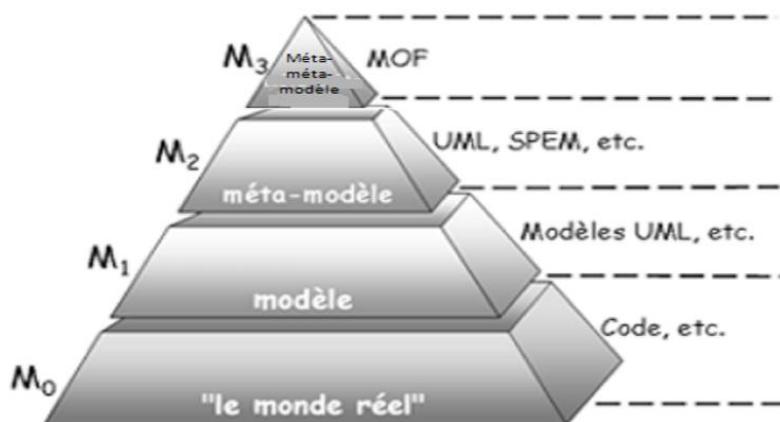


Figure 18: Niveaux de modélisation selon l'OMG (Bézivin, 2003)

L'OMG est un consortium sans buts lucratifs fondée en 1989 et regroupant des industriels, des fournisseurs d'outils, et des académiques dont l'objectif principal est de développer des standards pour l'industrie informatique. L'OMG est notamment à la base des standards UML (Unified Modelling Language), MOF (Meta-Object Facility), CORBA (Common Object Request Broker Architecture) et IDL (Interface Definition Language). Ces standards sur lesquels repose l'IDM sont centrés sur les notions de métamodèles et de méta-métamodèles et se trouvent sur différents niveaux dans l'architecture à quatre niveaux de l'OMG. Dans cette architecture (figure 18), le monde réel est représenté à la base de la pyramide (niveau M0). Au niveau M1 cette réalité est « modélisée » via un langage formel permettant de la décrire (modèle). Au niveau M2, un métamodèle permet de définir le langage que nous utilisons au niveau M1. Enfin, nous retrouvons au niveau M3, le standard MOF définissant un méta-métamodèle permettant de définir les métamodèles des langages. Le standard MOF s'auto-décrit limitant ainsi l'architecture pyramidale de l'OMG à quatre niveaux.

En résumé, L'IDM est une approche de conception tissant des liens entre métier et technique. Cette approche recentre l'ingénierie non pas sur le code source mais sur des modèles afin de simplifier les processus de maintenance et d'interopérabilité des systèmes. Tout l'enjeu repose donc sur les procédés de transformation de ces modèles. Cet enjeu donne lieu à d'importants travaux aussi bien au niveau industriel que académique afin d'une part d'enrichir les modèles et métamodèle de conception et d'autre part de fournir des outils informatiques

automatisant au maximum les transformations. Le succès d'UML en génie logiciel encourage l'ingénierie dirigée par les modèles et l'OMG tente aujourd'hui d'aller plus loin avec la gestion des processus métiers (BPM). Cette dernière s'aventure sur le terrain de la modélisation d'activité humaine en interaction avec la machine qui sera au cœur de notre chapitre 4. Avec l'expansion du web 2.0 l'interaction homme-machine se déroule souvent par le biais de services. L'IDM se couple alors à une autre approche du génie logiciel nommée SOA (service oriented architecture) afin de développer des applications modulaires et interconnectables. BPM, IDM et SOA forment ainsi un paradigme de conception transformant l'idée du « *conçu pour durer* » au « *conçu pour changer* » (Occello, 2002).

5.2 Architecture orientée services.

L'architecture orientée services (SOA) se compose d'un ensemble de méthodes pour le développement et l'intégration de systèmes dont les fonctionnalités sont développées sous forme de services interopérables et indépendants (Dumez, 2010). Une infrastructure SOA vise à permettre l'échange d'informations entre applications dont la cohérence interne est assurée par l'utilisation d'un format d'échange pivot (le plus souvent XML). Un service peut être défini comme un « *composant logiciel ou et/ou matériel capable d'agir avec précision afin d'accomplir des tâches pour le compte de l'utilisateur* » (Nwana, 1996, p.5). Ces services sont la plupart du temps des composants accessibles via l'utilisation d'un réseau tel que internet. Le terme « service web » qualifie alors tout service disponible par internet qui utilise un format standard d'échange de messages et qui « *n'est pas lié à un système d'exploitation ou un langage de programmation particulier* » (Dumez, 2010, p.15).

Les services web sont le plus souvent des interfaces de programmation¹⁸ (API) fournissant une porte d'accès à une fonctionnalité informatique utilisable par incorporation dans de nombreux programmes. Le fonctionnement de l'API représente le minimum qu'un informaticien ait besoin de connaître pour pouvoir assembler le composant à son application. Même si le fonctionnement interne du composant est hors du domaine de compétence de l'utilisateur, ce dernier peut s'en servir en sachant comment se nomme le composant, comment l'appeler et le faire interagir avec son système. L'API peut être vu comme une sorte de « boîte à outils » destinée aux développeurs leur permettant d'accéder aux données et fonctionnalités de façon sécurisée et contrôlée. La plupart du temps le processus est le suivant :

1. Une équipe informatique, une entreprise etc. veut mettre à disposition à d'autres développeurs leur données/fonctionnalités.
2. Elle construit une interface de programmation API pour contrôler et sécuriser cette utilisation.
3. Une fois l'API mise en place les développeurs tiers s'identifient, demandent un accès et peuvent faire usage des outils mis à disposition pour utiliser ces données et fonctionnalités dans leur propre application.

Afin d'assurer l'interopérabilité et la communication entre applications, les services web ont besoin d'être très standardisé et d'utiliser des protocoles de communication. Sans entrer dans les détails nous présentons ci-dessous les langages de base permettant de comprendre l'architecture et le fonctionnement des services web.

WSDL: Web Services Description Language est un langage de description basé sur XML permettant la description de service web. Le WSDL décrit une Interface publique d'accès à un Service Web, notamment dans le cadre d'architectures de type SOA. La version actuelle du WSDL (version 2.0) est approuvée et recommandée par le W3C. Un document WDSL contient quatre sections principales (Visible en jaune sur l'exemple illustratif donné en annexe 2).

1. *La section interface* définit les fonctions fournies par le service.

¹⁸ Un logiciel offrant des services à d'autres logiciels. Une API met l'accent sur les fonctionnalités en cachant les détails de son fonctionnement.

2. La section *types* définit en XML-schéma les types de données utilisés pour chaque message de type requête ou réponse.
3. La section *binding* indique le protocole de communication à utiliser, généralement SOAP (voir ci-après).
4. La section *service* définit le nom du service ainsi que l'adresse du service afin de pouvoir l'invoquer.

Le WSDL est indépendant de tout langage de programmation ou plateforme mais est directement interprétable par la machine. Il est même possible d'automatiser ce processus afin d'intégrer de nouveaux services en écrivant peu ou pas de code manuellement (ibid., p.18).

SOAP: Simple Object Access Protocol intervient au niveau des échanges entre services web. C'est un protocole de RPC (Remote Procedure Call) orienté objet bâti sur XML. Un RPC est un protocole réseau permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications pour assurer la communication entre le client, le serveur et des éventuels intermédiaires. SOAP permet donc la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP ou HTTPS (chiffrement) mais peut également se faire par d'autres protocoles comme SMTP. SOAP est actuellement le standard recommandé par le W3C. Il offre l'avantage d'être ouvert et extensible, de s'adapter à différents protocoles de transport, et tout comme WSDL d'être indépendant des langages et plateformes. En contrepartie il peut se révéler assez lourd du à la longueur des fichiers XML.

UDDI: Universal Description Discovery and Integration est une spécification technique sponsorisée pour la description, la découverte et l'intégration de services Web. UDDI constitue une partie importante de la pile des services Web, permettant aux personnes de publier et rechercher des services Web. Un annuaire UDDI permet de localiser sur le réseau le service Web recherché. UDDI permet une structuration de donnée ainsi que la spécification d'une API implémentée sous forme de service Web SOAP. L'annuaire UDDI se concentre sur le processus de découverte de l'architecture orientée services (SOA), et utilise des technologies standards telles que XML, SOAP et WSDL. Les données du registre UDDI sont stockées en trois catégories.

- Les pages blanches comprennent la liste des entreprises ainsi que des informations associées à ces dernières (coordonnées, description de l'entreprise, identifiants...).
- Les pages jaunes recensent les services Web de chacune des entreprises sous le standard WSDL.
- Les pages vertes fournissent des informations techniques précises sur les services fournis.

Ci-dessous nous représentons schématiquement l'architecture standard d'un service Web selon les trois spécifications présentées.

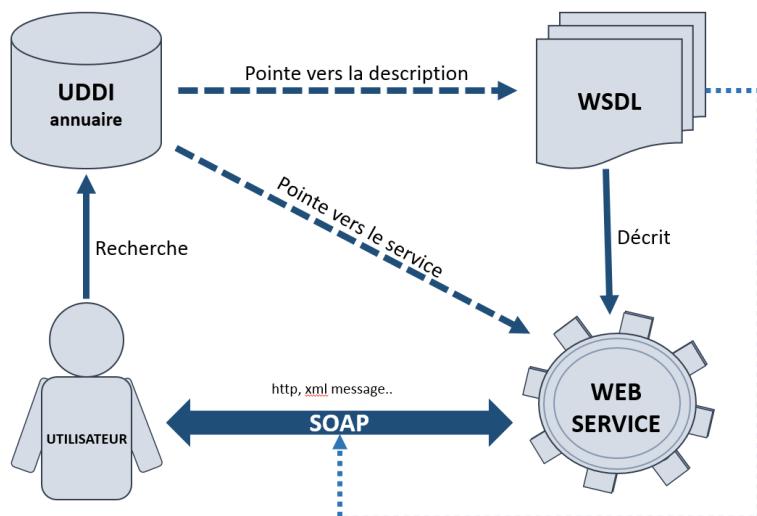


Figure 19: Représentation schématique de l'architecture d'un service web selon l'approche SOA

5.3 Enjeux pour les EIAH.

Les avancées de la recherche dans le champ de l'IDM permettent d'envisager de nouveaux mécanismes visant à proposer des environnements métiers accessibles et aptes à être opérationnalisés de façon informatique. Soutenu par des organismes influant (OMG, W3C) et des industriels incontournables dans le domaine informatique (IBM, ORACLE, Microsoft etc.) l'ingénierie dirigée par les modèles semble être l'orientation choisie pour le développement logiciel des prochaines années. Dans ce contexte, s'intéresser à l'IDM pour l'ingénierie des EIAH n'est pas dénué de sens. D'autant plus lorsque l'IDM partage certaines des préoccupations de l'ingénierie pédagogique lorsque celle-ci s'intéresse à la modélisation des scénarios d'apprentissage. Le développement d'environnements informatiques permettant la modélisation, le partage, l'exécution et la réingénierie de scénario pédagogique doit pouvoir s'inspirer de l'IDM. De plus comme le constate De Moura Filho & Derycke (2007) les avancées récentes en IDM permettent d'entrevoir un processus global de développement informatiques pouvant être « *raccourci et facilité, au point de permettre à des non-informaticiens – par exemple des concepteurs pédagogues – de pouvoir contribuer de manière significative à l'effort de développement, dans un nouveau processus que Fowler (2005) appelle le Lay Programming.* » (p.120). Si la participation d'un enseignant reste encore utopique, l'introduction croissante des intentions pédagogiques dans les processus de conception des EIAH par l'intermédiaire de « nouveaux » acteurs tels que les ingénieurs pédagogiques, paraît envisageable.

Le paradigme d'Architecture Orientée Services quant à lui représente de même un enjeu de taille pour l'ingénierie des EIAH. Nous pensons en effet, qu'avec le processus de conception inverse (« monolithique ») de ces derniers, le pédagogue est « prisonnier » des limites de l'environnement qu'il utilise. Aussi polyvalent soit-il les environnements informatiques cristallisent en eux des usages prédéfinis pouvant dans certains cas aller à l'encontre des intentions pédagogiques. Il n'est pas rare d'observer des dispositifs e-learning entièrement basés sur les fonctionnalités offertes par l'environnement informatique mise en œuvre au sein de l'institution. Ce ne sont plus les intentions pédagogiques qui dirigent la construction d'un dispositif mais l'artefact. En outre, nous ne pensons pas que la recherche de polyvalence toujours plus grande des EIAH soit une solution pour limiter ce risque. En effet il est possible de constater « un paradoxe » concernant la polyvalence des artefacts informatiques. La volonté d'offrir toujours plus de fonctionnalités entraîne bien souvent l'accroissement de la complexité de l'outil informatique. Hors plus un EIAH devient complexe plus les possibilités de détournement (principe de catachrèse de Rabardel) de l'usage de ce dernier diminue. Les praticiens ne sont alors plus en mesure d'utiliser l'outil autrement que ce pour lequel il a été construit.

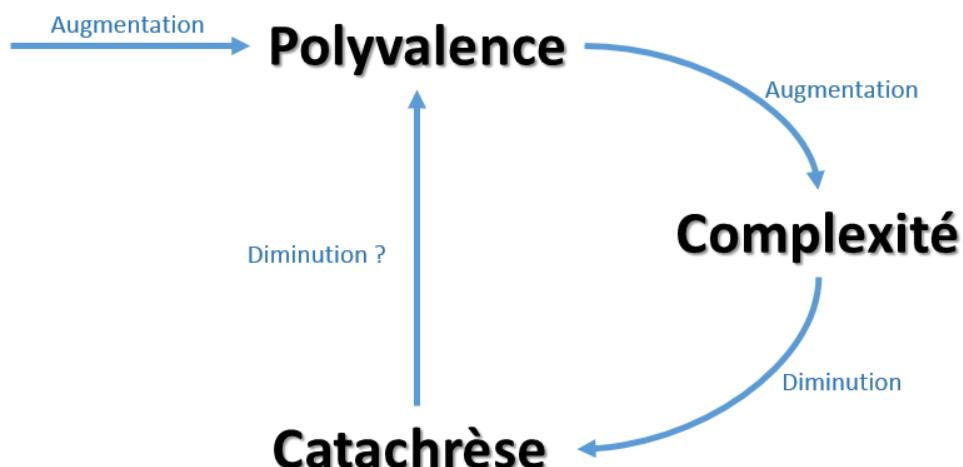


Figure 20: Paradoxe de la polyvalence des EIAH

Le « paradoxe de polyvalence » est selon nous complémentaire d'un autre paradoxe concernant la conception des EIAH identifié cette fois-ci par Carron (2007) : « *un dispositif pédagogique n'est pas prévu pour un EIAH spécifique* » (p. 31). Ce paradoxe s'appuie selon l'auteur sur trois idées :

1. *L'étrangeté* : Dans la mesure où l'équipe d'enseignants chargée d'utiliser l'EIAH n'a pas participé à sa réalisation on ne peut pas parler de véritable adéquation entre celui-ci et les enseignants.
2. *La dualité* : Intention pédagogique et EIAH forment un couple dont chaque partie modifie l'autre. Autant l'utilisation d'un EIAH permet de faire évoluer les intentions pédagogiques d'une équipe d'enseignants, autant l'EIAH se finalise dans l'usage que l'équipe en fait.
3. *La différence de cycle* : Création pédagogique et création d'EIAH appartiennent à deux cycles de création différents dont la synchronisation nécessite des outils de négociation.

Dans ce contexte il est peut être intéressant de réfléchir aux principes d'une SOA. En effet, cette dernière propose un environnement flexible construit à partir des services nécessaires à l'objectif visé. Le concepteur pense un dispositif pédagogique, recherche ensuite le ou les services qui lui semblent adéquat et les connecte à son EIAH. Cependant en pratique pour qu'un tel dispositif fonctionne nous recensons deux impératifs :

- *L'abondance* : l'offre de service doit être suffisamment large pour couvrir les besoins des dispositifs pédagogiques les plus innovants.
- *La découverte* : Ce principe signifie qu'un acteur peut facilement savoir qu'un service existe et savoir comment l'utiliser. « *Même si un service fournit une fonctionnalité importante, il serait très inefficace s'il n'était pas découvrable pour être réutilisé plus tard.* » (Dumez, 2010, p.14)

Les solutions de web service se multiplient dans le « e-business » et les technologies de catalogage tel que UDDI participent activement au « principe de découverte » chère à la SOA. Cependant il reste encore beaucoup à faire

dans le domaine de l'éducation. De plus, Boudali, Balla & Amrouche (2009) constate que le standards UDDI n'offre qu'une description syntaxique des services. Il propose d'introduire une technologie de web services sémantiques permettant d'améliorer la qualité de la découverte dans le cadre de web services orientée e-learning. Il présente un système de recherche de web services e-learning dans lequel UDDI est complété par une ontologie adaptée pour l'éducation. Les auteurs enrichissent l'ontologie générique DAML-S (Garner 2004) avec des classes décrivant des informations pédagogiques, techniques et financières pour l'utilisation du web service dans un EIAH. Le système proposée respecte ainsi les informations WSDL nécessaires à l'invocation du service et est compatible avec l'indexation

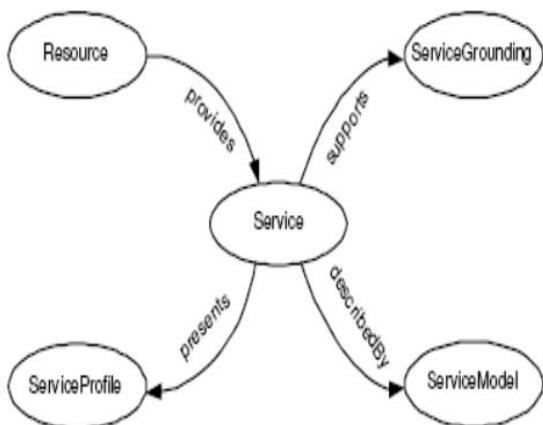


Figure 21: Classes principales ontologie DAML-S
(Garner, 2004)

UDDI. Une interface avec profil utilisateur et un algorithme de recherche permettant d'effectué des recherches filtrés viennent complétés le dispositif proposé. Si ce dernier n'est qu'un prototype, il laisse entrevoir les futures perspectives de la SOA pour l'ingénierie des EIAH.

CONCLUSION

Le développement de méthodes et technologies permettant de soutenir les intentions pédagogiques des enseignants sont au cœur de l'ingénierie pédagogique. Cette dernière ne se concentre pas uniquement sur l'interopérabilité des contenus de formation mais recentre son approche sur l'activité des apprenants et la scénarisation d'une équipe pédagogique. Que ce soit par la création de scripts CSCL formalisant les interactions en contextes collaboratifs ou le développement de standard permettant l'échange et l'opérationnalisation de scénarios, on constate une volonté de créer des liens facilitant le transfert des conceptions pédagogiques vers l'informatique et une volonté d'échange et de partage rationalisant les efforts de développement. Nous constatons alors un certain intérêt pour la modélisation.

Les différents travaux abordés au cours du chapitre mettent en exergue une diversité intéressante d'approches concernant le développement de langages de modélisation pédagogique (notamment des langages visuels). Cependant nous constatons que beaucoup d'entre eux restent au stade expérimental. Lorsque ces derniers sont en production, le manque d'outils de développement simples et efficaces entrave leur diffusion et leur utilisation. Lorsqu'un outil ne convient pas aux attentes peu d'alternatives s'offrent alors au concepteur. C'est un des points justifiant l'approche différente empruntée par ce mémoire, nous explorons la possibilité d'utiliser dans une visée éducative un langage de conception déjà existant.

Au cours de ce chapitre nous avons également mis en exergue le rôle d'ingénieur pédagogique de plus en plus présent au sein des développements de formations instrumentées. D'une part, assurer le lien entre les préoccupations techniques et pédagogiques des divers acteurs impliqués dans la conception des dispositifs techno-pédagogiques nécessite la présence d'un acteur « charnière » à même de comprendre les différentes perspectives. D'autre part, la complexité des dispositifs innovants de formation en rupture avec les dimensions traditionnelles de temps et de lieu et devant tenir compte des contraintes opérationnelles, techniques et budgétaires, nécessite un acteur à même d'utiliser des méthodes normalisés d'ingénierie. Dans cette perspective nous pensons qu'un langage de modélisation pédagogique pourrait soutenir la pratique de cet acteur. Pour ce faire ce langage devrait selon nous, [1] être compréhensible par des acteurs d'horizons professionnels différents, [2] être opérationnel (outils), [3] faciliter l'implémentation (rationalisation de l'effort de modélisation) et [4] être normalisé et lié à un standard (échange entre concepteur, durée de vie etc.).

Enfin l'ingénierie pédagogique nous a naturellement amené à nous intéresser à l'ingénierie des EIAH. À travers les possibilités offertes actuellement par l'approche IDM et les principes de l'architecture SOA, nous nous sommes alors interrogés sur la possibilité de concevoir des EIAH modulaires et interconnectables générés à partir de modèles capturant les intentions pédagogiques. L'idée est de fournir des dispositifs techniques réellement en accord avec le design pédagogique en réintroduisant le concepteur pédagogique au cœur de l'approche de développement. Ainsi la démarche ne serait pas d'adapter un scénario pédagogique aux fonctionnalités d'un EIAH mais adapter l'EIAH à un scénario pédagogique.

Les deux chapitres suivant aborderont notre proposition d'utilisation du langage BPMN 2.0. L'enjeu de ce mémoire est d'explorer cette technologie dont l'utilisation et le développement sont en forte croissance afin d'en extraire les possibles applications dans le cadre de la modélisation pédagogique et le développement d'environnements d'apprentissage centrés sur les scénarios. Nous commencerons par présenter ce langage au chapitre 4 et testerons sa mise en œuvre en contexte éducatif au cours du chapitre 5. Nous présenterons alors la manière dont nous avons utilisé celui-ci pour modéliser des scénarios pédagogiques et les exécuter sur une plateforme de gestion de workflow.

CHAPITRE IV. BPMN 2.0 - INTRODUCTION AU STANDARD

Dans la perspective de décrire « le workflow » d'une scénarisation pédagogique et faciliter la conception d'EIAH basée sur une approche IDM, nous nous intéressons dans ce mémoire à un langage graphique de conception nommé BPMN (Business Process Modelling and Notation). Si ce dernier est issu du « monde de l'entreprise », le langage est abstrait et pourrait possiblement être utilisé dans le domaine des technologies éducatives. Comme le constate Chinosi et Trombetta (2012) :

« BPMN est de facto le standard pour représenter d'une manière graphique très expressive les processus qui se produisent dans pratiquement tous les types d'organisation, des recettes de cuisine aux processus d'affectation du prix Nobel, de la gestion des incidents aux systèmes de vote électroniques, ou encore les procédures de réservation de voyages, pour n'en nommer que quelques-uns».

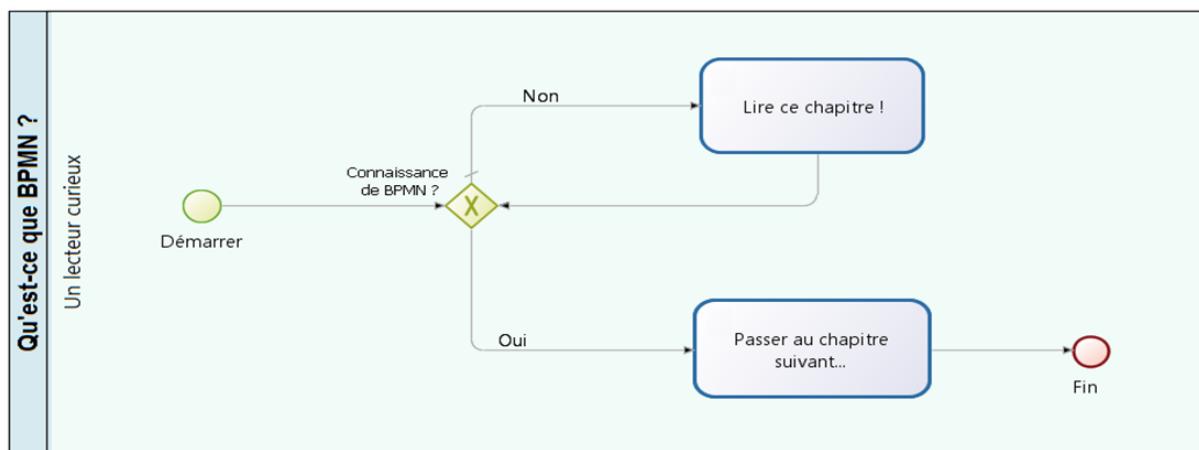


Figure 22: un diagramme BPMN purement illustratif, réalisé avec Bonita BPM.

INTRODUCTION

BPMN (Business Process Model and Notation) est une notation pour la représentation graphique des processus métier dans un workflow. BPMN est un langage récent conçu pour être au cœur d'une approche de modélisation et d'implémentation utilisant les cadres conceptuels de l'ingénierie dirigée par les modèles (IDM), de l'architecture orientée services (SOA¹⁹) et de la gestion des processus métier (BPM²⁰).

À l'origine BPMN est un projet initié par Business Process Management Initiative (BPMI) qui fusionna en 2005 avec l'Object Management Group (OMG). L'OMG est une association américaine à but non lucratif créée en 1989, dont l'objectif est de standardiser et promouvoir le modèle objet sous toutes ses formes. L'OMG est un organisme influant dans l'industrie de conception logiciel et est notamment à la base des standards UML (Unified Modelling Language), MOF (Meta-Object Facility) ou CORBA (Common Object Request Broker Architecture). Elle est aussi à l'origine de la recommandation MDA (Model Driven Architecture) développant les principes de l'ingénierie dirigée par les modèles (IDM que nous avons abordé au chapitre précédent).

Dans ce chapitre nous aborderons premièrement l'approche BPM qui sous-tend le langage et les apports de la nouvelle spécification 2.0. Nous présenterons ensuite brièvement le fonctionnement du langage (un chapitre complémentaire étant disponible en annexe). Enfin nous étudierons les différents outils de modélisation et environnements de développement utilisant le standard BPMN 2.0.

1. DE L'APPROCHE BPM AU STANDARD BPMN 2.0

1.1 Ingénierie des processus métier (BPM).

L'ingénierie des processus d'entreprise – ou métier – est souvent représenté par l'acronyme BPM signifiant Business Process Management. L'approche BPM trouve son origine dans les travaux sur le « Workflow », apparu avec l'industrialisation du secteur tertiaire, puis considérablement développé dans les années 90 avec leur gestion par systèmes informatiques (Ulmer, 2011). Un Workflow peut être défini selon la Workflow Management Coalition (1999) comme « *l'automatisation des procédures métiers²¹, en partie ou totalité, durant laquelle les documents, informations et tâches sont transférés d'un participant à un autre selon un système de règles établies* » (p.8). Aujourd'hui, le BPM correspond à une ingénierie des processus métier à l'aide des technologies de l'information. Il a pour vocation de modéliser, déployer, exécuter et optimiser de manière continue les différents types de processus et ainsi d'améliorer l'agilité d'une organisation (Ulmer 2011, p.63). Le BPM est devenu une pierre centrale de la construction d'applications logicielles, « *de la même façon qu'il faut définir le modèle de données, il faut désormais définir le modèle de processus des applications métier.* » (Debauche & Megard, 2004, p.14).

L'approche BPM définit toujours le cycle de vie d'un processus métier comme une boucle continue. L'essence d'une approche BPM est le principe de réingénierie. La modélisation d'un processus incluant l'activité humaine est souvent imparfaite mais doit pouvoir être évaluée et améliorée continuellement pour parvenir à la gestion optimale des activités. Nous présentons page suivante quelques représentations de ce cycle circulaire et incrémental.

¹⁹ Services Oriented Architecture

²⁰ Business Process Management

²¹ Les processus métier ont pour fonction d'orchestrer les activités incluant une interaction entre différents acteurs participant à l'accomplissement d'un but ou objectif de l'entreprise.

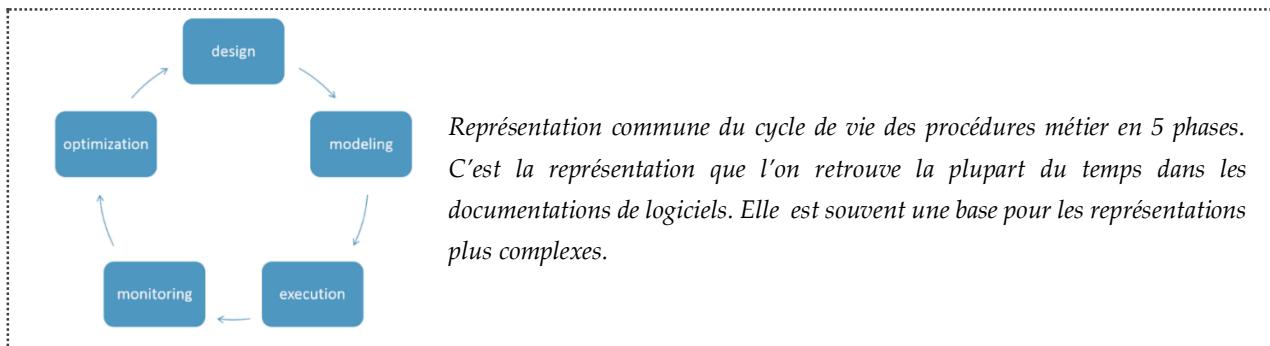


Figure 23: Représentation commune d'un cycle de vie des procédures métier

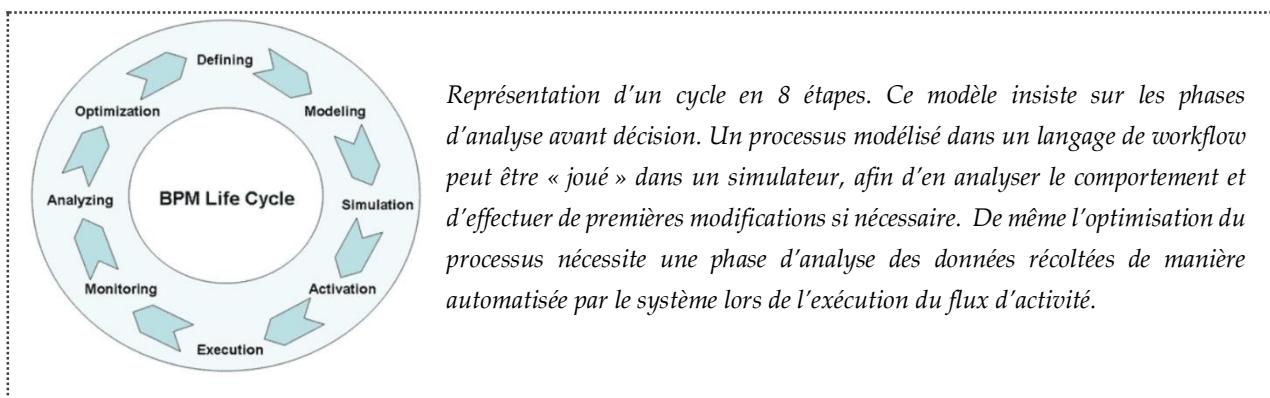


Figure 24: Cycle de vie des procédures métier (Gartner 2005)

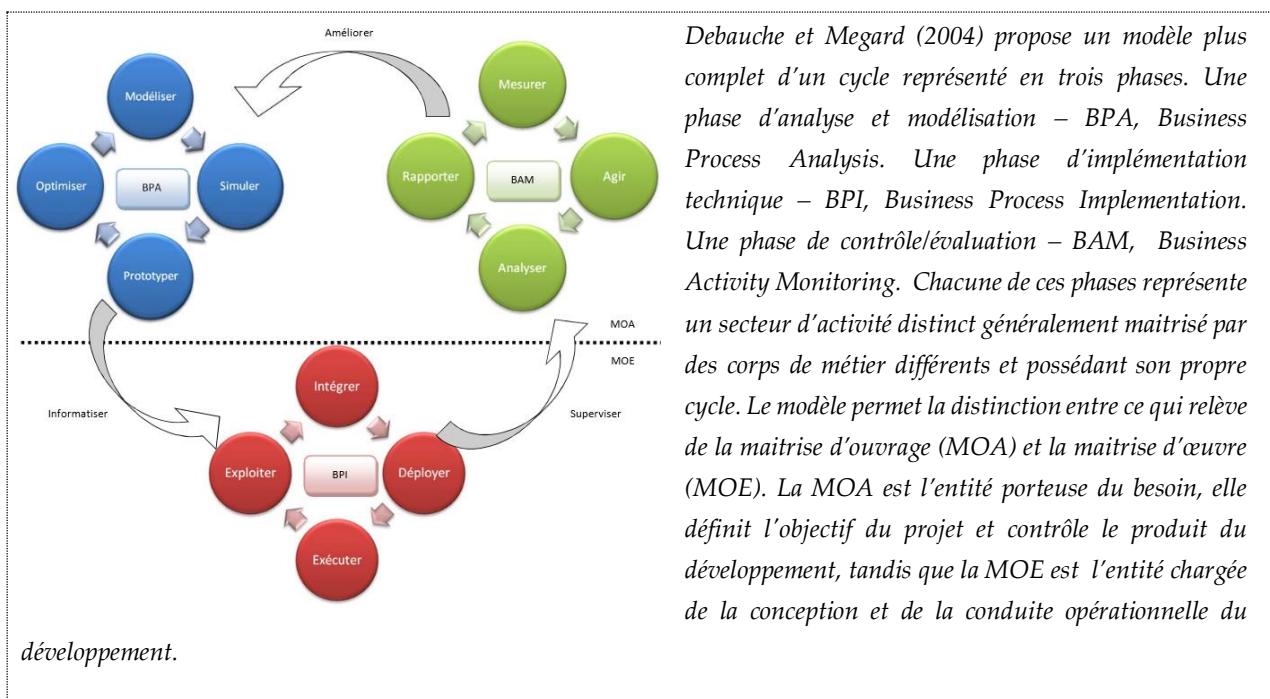


Figure 25: BPM - cycle de vie des processus métier (Debauche & Megard, 2004)

1.2 BPMN – un standard international pour l'ingénierie des processus.

La mise en œuvre d'une ingénierie des processus métier (voir point précédent) nécessita l'utilisation d'une technologie de soutien facilitant la modélisation, l'exécution et l'optimisation des procédures. La spécification Business Process Model and Notation a récemment émergé comme une norme internationale pour la modélisation des processus métier. C'est l'un des développements les plus importants dans l'histoire du Business Process Management et BPMN et il a le potentiel d'unifier les pratiques BPM de la même manière qu'UML a unifié le génie logiciel (Moody, 2012, p.1). L'objectif de BPMN est selon L'OMG :

« de fournir une notation qui est facilement compréhensible par tous les utilisateurs professionnels, des analystes métier qui créent la version initiale du processus, aux développeurs techniques chargés de l'application de la technologie qui va exécuter ces processus, et finalement, les personnes qui permettront de gérer et de contrôler ces processus. Ainsi, BPMN crée un pont standardisé pour l'écart entre la conception des processus d'affaires et l'implémentation des processus. » (OMG BPMN v.2.0.1, 2013, p.1)

Fin 2008, BPMN a atteint un « tournant » dans son évolution et il peut être considéré comme un standard massivement adopté (Silver, 2009). Jamais un standard n'avait été aussi largement adopté si rapidement après sa mise en place (Recker, 2010). Les raisons pouvant être avancées pour expliquer ce succès sont :

- BPMN est le premier standard international pour la modélisation de processus métier et il est supporté par un organisme dont la réputation dans le domaine de l'ingénierie n'est plus à faire.
- Il est adopté en pratique par des acteurs majeurs de l'industrie IT (IBM, SAP, Oracle.)
- La vision de BPMN recoupe complètement celle de l'Ingénierie dirigée par modèle (IDM) en pleine expansion dans l'univers du génie logiciel
- L'avenir de BPMN semble assuré et la définition d'une sémantique d'exécution dans la dernière version (BPMN 2.0) nourrit de nombreux espoirs de développement et d'interopérabilité.

1.3 De BPMN 1.2 à BPMN 2.0.

BPMN 1.2 était une notation graphique non exécutable. Des solutions plus ou moins formelles se sont développées pour implémenter les modèles BPMN via des systèmes de conversion vers d'autres langages exécutables. La plupart du temps les solutions consistaient à passer par le langage Business Process Execution Language (BPEL). Aujourd'hui encore plusieurs outils commerciaux proposent de compiler les diagrammes vers un BPEL exécutable. Cependant cette solution est partielle et souvent peut opérationnelle et non standardisé. Certaines modélisations BPMN ne peuvent être converties, notamment car la définition des tâches humaines devient problématique, et il faut souvent passer par un « mapping » manuel de BPMN vers BPEL.

En janvier 2011, l'OMG présenta BPMN 2.0 comme successeur à BPMN 1.2. BPMN 2.0 évolue vers un schéma d'échange standard en XML permettant l'échange de modèles exécutables, en remplacement de BPEL. Selon Chinosi et Trombetta (2012), la spécification étend la portée et les capacités de le BPMN 1.2 dans plusieurs domaines :

- elle officialise l'exécution sémantique pour tous les éléments BPMN
- elle définit un mécanisme d'extensibilité du langage
- affine la composition et la corrélation d'événements
- étend la définition des interactions humaines
- définit des nouveaux diagrammes de chorégraphie et de conversation
- résout les incohérences et les ambiguïtés de BPMN 1.2

De manière générale BPMN 2.0 est un langage graphique standardisé et basé sur une structure XML. Il permet de dessiner des diagrammes représentant le déroulement et l'enchaînement de tâches. À l'aide de différents éléments standards, une séquence d'activités d'un processus peut être détaillée en vue de l'atteinte d'un objectif.

BPMN est spécialement indiqué pour la représentation d'activité humaine en interaction avec un dispositif informatique quelconque, toutefois il peut aussi être utilisé pour représenter des activités n'impliquant pas d'interaction homme-machine.

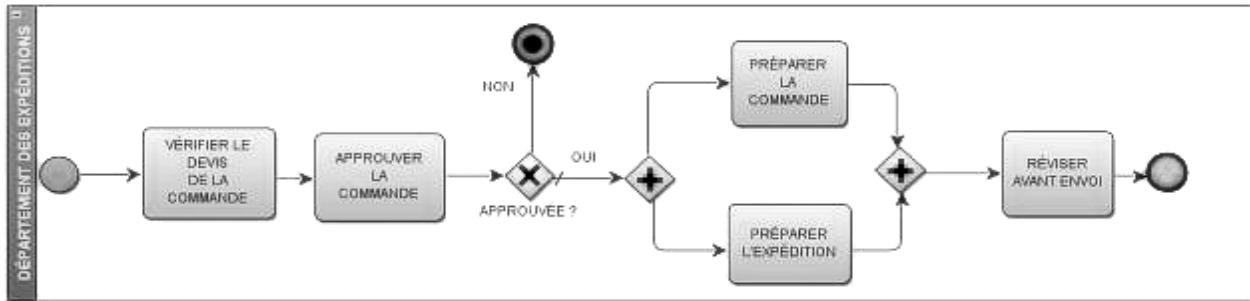


Figure 26: Processus d'expédition des commandes avec BPMN 2.0 (OMG, 2013, p.170)

Un diagramme BPMN peut être modélisé via un outil de modélisation BPMN nous indiquant les erreurs et permettant l'export dans un format standard en XML. Pour le développement d'application, il existe des outils plus puissants, nommés BPMS (Business Process and Management System) permettant de :

- simuler le comportement de notre processus,
- connecter les activités à des services informatiques
- exécuter notre application via un moteur de workflow
- déployer notre application sur une plateforme web pour gérer les utilisateurs
- générer des rapports de données pour l'analyse et l'amélioration de notre processus

2. BPMN 2.0

Un chapitre est disponible en annexe proposant une présentation plus complète des éléments du langage et son fonctionnement. Nous introduirons ici les bases « suffisante » pour la lecture de ce mémoire.

BPMN est constitué d'un ensemble d'éléments de modélisation. Ces éléments sont constitués d'un symbole (objet graphique sur un diagramme) et d'une liste d'attributs (la plupart du temps invisible sur un diagramme). Remplir les attributs associés à un élément est généralement nécessaire dans une perspective d'implémentation. Les symboles graphiques sont nombreux (environnées une centaine). Cependant ce sont en grande majorité des variantes des représentations graphiques basiques organisées en quatre catégories (figure 27). Les éléments décrits par la norme se différencient par leurs formes et l'épaisseur, la continuité ou la discontinuité de leurs traits. Aucun attribut de couleur n'est défini dans la norme. Cependant pour des raisons esthétiques et/ou ergonomiques les logiciels de modélisation présentent des versions colorées de ces symboles ou proposent une palette de couleur pour les colorer de manière personnelle. Ainsi, représenter les éléments « de début » en vert et les éléments « de fin » en rouge est une règle informelle fréquemment adoptée par les outils de modélisation.

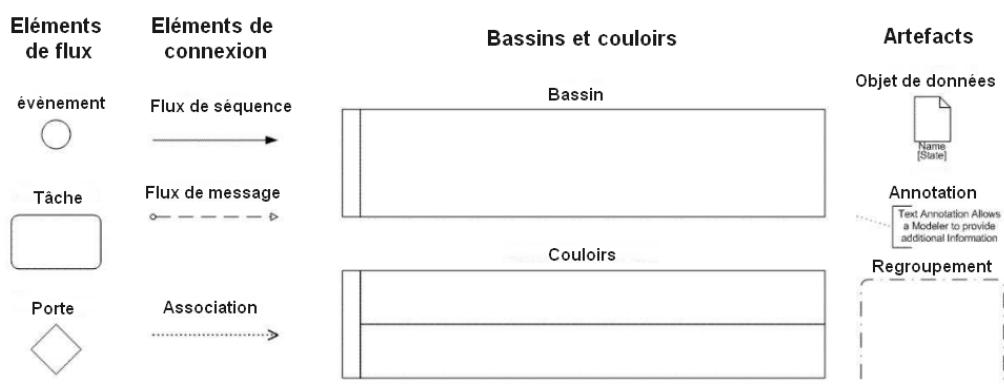


Figure 27: éléments principaux

De manière simple, les tâches représentent une action effectuée par un humain ou un service informatique. Ces actions sont liées entre elles par des flèches de connexion, nommées flux de séquence, qui représentent le cours du processus, son « flux ». Les portes représentent des points de jonction où le flux se divise en chemins parallèles ou conditionnels. Enfin des « événements » représentent toute chose se produisant et influençant le déroulement du flux (début, pause, fin, arrêt, erreur, attente de signal etc.). Tous ces éléments sont ensuite placés dans des bassins représentants les limites du processus. Ce bassin est divisé en couloirs représentant les acteurs du dispositif. Nous pouvons voir qui effectue quoi en regardant dans quel couloir se situe la tâche. Ce fonctionnement est illustré par la figure suivante.

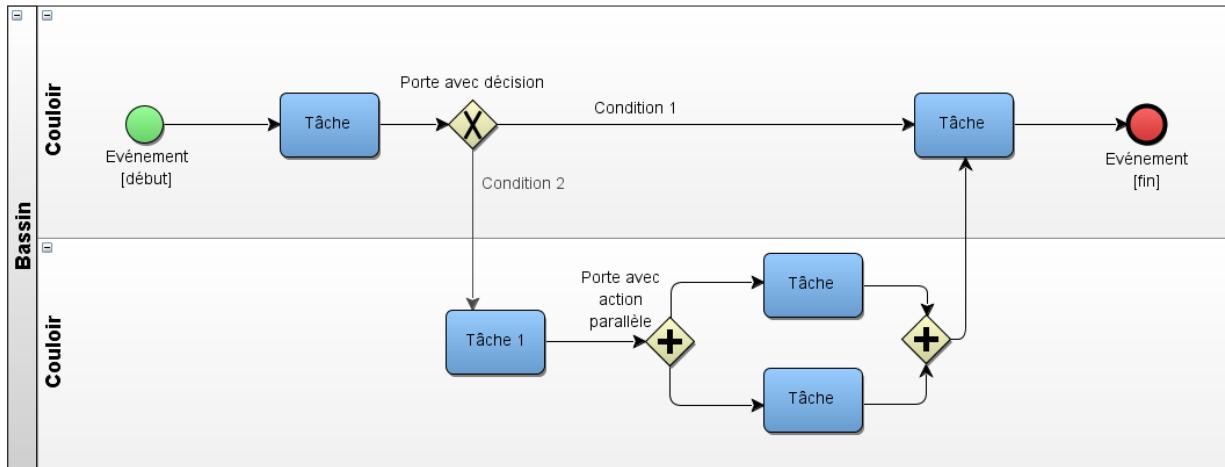


Figure 28: Exemple illustrant le fonctionnement de BPMN

L'exemple précédent est constitué d'un seul bassin, donc un seul processus avec un début et une fin. Différents « bassins » représentent différents processus indépendants. Une connexion « flux de séquence » ne peut jamais sortir des limites d'un bassin. Cependant différents processus peuvent communiquer « via des flux de message » représentant le parcours des communications.

Lorsqu'un seul processus (un seul bassin) est représenté le diagramme est nommé processus privé. Lorsqu'un processus détaillé communique avec un autre processus non détaillé (un simple bassin vide) le diagramme est nommé processus public. Enfin lorsque deux processus détaillés communiquent, le diagramme est nommé « collaboration ». Ce sont ces diagrammes que nous aborderons dans ce mémoire.

Il existe 2 autres types de diagramme définis par la norme de l'OMG : Les diagrammes de « chorégraphie » et de « conversation ». Ces deux derniers diagrammes sont nouveaux et ont seulement été introduits dans la spécification BPMN 2.0. Nous les présentons en annexe mais ne les utiliseront pas dans ce mémoire pour les raisons suivantes : Ils ne sont encore pas gérés, ou incomplètement gérés, par les outils de modélisation actuels. Leur utilisation par les professionnels est encore rare. Nous nous centrons dans ce mémoire sur l'implémentation d'activités pédagogiques orchestrés.

Toutefois en axant la modélisation sur les processus communicationnels leur utilisation pourrait représenter un sujet d'investigation intéressant dans le cadre d'une approche CSCL.

3. OUTILS ET PRINCIPES DE DÉVELOPPEMENT

Contrairement aux langages de modélisations pédagogiques dont les outils manquent, BPMN est soutenu par une industrie importante développant de nombreuses solutions logicielles. Ces outils se distinguent premièrement par leur niveau de conformité au langage. La norme 2.0 (OMG, 2013) définit quatre types de conformité à savoir :

1. Conformité de modélisation de processus / Process Modelling Conformance,
2. Conformité d'exécution de processus / Process Execution Conformance,
3. Conformité d'exécution BPEL / BPEL Process Execution Conformance
4. Conformité de modélisation processus de type chorégraphies/ Choreography Modeling Conformance.

Pour chacun de ces 4 points la norme définit un certain nombre de spécifications que tout logiciel doit impérativement remplir s'il veut se réclamer de l'un ou de l'ensemble des 4 types de conformité BPMN 2.0. Le premier niveau concerne surtout les outils de modélisation simples permettant de dessiner des diagrammes compatibles BPMN à un niveau d'abstraction assez élevé. Les outils de modélisation avancés respecteront aussi souvent le niveau de conformité 2 permettant d'utiliser beaucoup plus d'éléments et d'attributs d'éléments afin de dessiner des diagrammes de plus bas niveaux pour l'implémentation. Le niveau 3 est un complément du 2 et est surtout destiné aux anciens outils utilisant une conversion vers le langage BPEL pour l'exécution. Enfin le niveau 4 concerne la possibilité de dessiner des diagrammes de type « Chorégraphie ». En pratique il est souvent difficile de savoir à quel niveau de conformité se situe un outil. Les documentations logicielles sont parfois floues sur ce point où ne l'abordent pas. La plupart du temps les fournisseurs présentent leurs stratégies de développement, la vision dans laquelle leurs produits s'inscrivent, sans forcément faire référence à la norme officielle BPMN 2.0. Toutefois on remarque une amélioration récente sur ce point où la concurrence et la pression effectuée par des utilisateurs œuvrent pour plus de transparence.

Les outils se distinguent ensuite par leur finalité. Certains ne visent que la modélisation, d'autre se veulent des environnements de développement applicatif complet. On nomme ces derniers des BPMS (Business Process Management System). Les BPMS sont en général des suites logicielles permettant de modéliser, exécuter et gérer des applications en ligne. Ces suites se composent :

- D'un outil de modélisation qui servira à formaliser la description de processus d'interactions homme-machine avec le langage BPMN. La modélisation nécessite de compléter des attributs d'éléments permettant notamment de définir les données associées et des règles conditionnelles.
- D'une interface de développement permettant de lier des services informatique aux diagrammes via des API.
- D'un moteur d'exécution s'occupant du transfert vers le code machine.
- D'un moteur de règles qui évaluera l'état de tous les objets impliqués dans le déroulement des processus et déterminera si les conditions sont remplies pour lancer, poursuivre ou arrêter l'exécution.
- D'un référentiel (base de données, serveurs) qui mémorisera tous les objets manipulés, en particulier les définitions des processus, les règles qui doivent déclencher leur exécution, les contraintes d'intégrité, de sécurité ainsi que les mesures de référence relatives au métier de l'entreprise.
- D'un outil d'administration (la plupart du temps un environnement en ligne) permettant de régler l'ensemble du système (notamment concernant la gestion des utilisateurs). Ces outils d'administration permettent systématiquement d'obtenir des indicateurs de performance et des statistiques à partir des données collectées lors de l'exécution des processus. Cette fonctionnalité est au cœur de l'approche BPM et son processus de réingénierie.

3.1 Outils de modélisation.

De manière classique un outil de modélisation BPMN se compose :

- D'un espace de dessin central permettant la modélisation.
- D'une palette d'éléments graphiques que l'on déplace dans l'espace de dessin par « drag & drop ». ces éléments peuvent parfois être colorés de manière personnalisée via une palette graphique de couleurs.

On retrouve ensuite les fonctionnalités suivantes :

- Un moyen de remplir les attributs accompagnant les éléments graphiques. En général cette manipulation s'effectue soit par un clic droit sur l'élément choisi soit par l'intermédiaire d'un panneau contextuel.
- Des outils de validation de la conformité du diagramme à la norme 2.0. Les validations s'effectuent soit en direct (validation live), soit à posteriori en cliquant sur un bouton lançant une recherche d'erreur ou lors de l'export du diagramme.
- Un menu d'export/import des diagrammes. Il existe pour cette fonctionnalité de grandes différences entre les outils. A minima l'export et l'import de fichier BPMN 2.0 doit être présent (ce n'est pas toujours le cas !)

Enfin les outils plus avancées proposent généralement :

- Des outils de simulations du processus permettant de vérifier son fonctionnement avec des utilisateurs fictifs.
- Des fonctions intégrées de partage avec la communauté.
- De nombreux formats d'exports (PNG, PDF, Mediawiki, résumé des attributs au format XML, auto-génération de documentation pour développeur, conversion totale ou partielle vers d'autres langages)

S'il existe plus de 60 outils pour la modélisation BPMN selon l'OMG leur qualité est très diverse. Le choix d'un modélisateur BPMN s'effectue sur :

- Son respect du standard. Niveau de conformité avec BPMN 2.0 (respecter à minima les types de conformité niveau 1 et 2).
- Son « look & feel » basé sur des critères d'ergonomie, de simplicité d'utilisation et de réactivité de l'application.
- Sa plateforme d'utilisation (Windows, Linux, Mac), le choix de pouvoir être utilisé directement en ligne (cloud) ou hors-ligne (standalone).
- Son statut commercial, gratuit, open-source etc.
- La puissance de ces fonctionnalités (validateur de diagramme, simulateur)
- Son interopérabilité. Les fonctionnalités d'import/export des diagrammes en différents formats.

Nous présentons en *annexe 6*, des modélisateurs gratuits respectant le standard BPMN 2.0 et ne faisant pas partie d'un BPMS (ou pouvant en être séparé). Dans le cadre de ce mémoire nous utilisons le logiciel Yaoqiang BPMN Editor 2.2. Ce dernier est léger, gratuit et open-source et respecte en intégralité la spécification 2.0. Il est à noter qu'il existe en plus, des extensions logicielles permettant de modéliser des diagrammes BPMN dans des environnements plus génériques. Certaines sont de bonne facture et représentent une solution efficace de modéliser avec BPMN. Citons par exemple « BPMN Visio Modeller » une extension permettant de dessiner des diagrammes conforme à BPMN 2.0 dans le logiciel Visio de la suite bureautique Microsoft ou encore les différentes extensions permettant de modéliser des diagrammes avec l'IDE Éclipse.

3.2 Outil de développement complet (BPMS).

Comme nous l'avons dit les BPMS sont des suites logicielles complètes pour le développement et la gestion d'applications orientées processus. Elles intègrent généralement des outils de modélisation couplés à des API de développement, un moteur de workflow et une plateforme d'administration pour analyser et gérer le déroulement des processus. Ces solutions sont connues sous le nom de Business Process Management System (BPMS) et suivent les différentes étapes du BPM Life cycle que nous avons abordé précédemment.

Les outils BPMS sont impliqués dans un très grand nombre de développements applicatifs de grande ampleur au sein d'institution gouvernementale et de multinationales. Aussi la plupart des suites sont commerciales et évoluent sur un marché très concurrentiel. Les BPMS sont généralement accompagnés d'une équipe de support et de développement importante pour accompagner les clients utilisant la solution dans un projet de développement.

De manière générale, deux approches commerciales semblent se distinguer dans l'offre d'outil des BPMS :

- *L'approche "Zero-coding"* signifie qu'un modèle de processus graphique peut être rendu exécutable à l'aide de blocs de construction prédéfinis et de boîtes de dialogue de configuration. Les blocs de construction (services, adaptateurs, composants de transformation, etc.) couvrent de nombreux scénarios typiques et peuvent être orchestrés sous la forme du modèle de processus. Cela nécessite des mappages de données afin de relier les entrées et sorties de tâches. Cette approche est très efficace lorsque les éléments constitutifs couvrent tous les besoins. Cependant les outils sont encore peu nombreux et le « zéro-code » est un argument marketing rarement observable en pratique. Toutefois, en limitant au maximum le temps de programmation elle accélère considérablement les cycles de production d'applications métier fonctionnelles. La demande est forte pour ce genre de logiciel permettant à des analystes métiers de tous domaines mais non-informaticien de participer activement aux processus de conception logiciel. La norme BPMN 2.0 est encore très récente et il n'est pas impossible que des solutions réellement utilisables par des personnes non techniques voient le jour. Bonita BPM, Sydle Seed, Intalio BMPS, ProcessMaker sont des suites logicielles se déclarant ouvertement tenant de cette approche.
- *L'approche "less-coding"* argumente quant à elle que les éléments constitutifs du logiciel ne peuvent pas couvrir tous les besoins. Comme le nombre de scénarios de réalisation est potentiellement infini, la configuration en "zéro-coding" des projets peut être très complexe. Pour les tenants de cette approche, la configuration "zero-coding" demande parfois plus d'effort que le développement logiciel classique. C'est pourquoi il considère qu'un compromis est préférable avec une application qui génère une partie du code que l'on complète par la suite. Cette approche est plutôt plébiscitée par les IT et demande des compétences en programmation (la plupart du temps en langage Java). L'approche "less-coding" est donc plus complexe pour un non initié mais aussi plus souple et puissante. Les suites logicielles tenant de cette approche promeuvent en général au maximum l'interopérabilité avec les IDE favoris des développeurs (Ex: Activiti et Éclipse). JPBPM, Activiti, Signavio ou Agility sont des exemples de BPMS inscrits dans cette approche.

Nous présentons en annexe 5 un tableau regroupant quelques BPMS célèbres offrant des solutions gratuites ou partiellement gratuites. Parmi ces offres de solution, notre choix s'est porté sur Bonita BPM 6 de la société BonitaSoft. Issu du projet d'une start-up française initié en 2001, la suite Bonita est open-source et propose une version Community gratuite. Elle se compose :

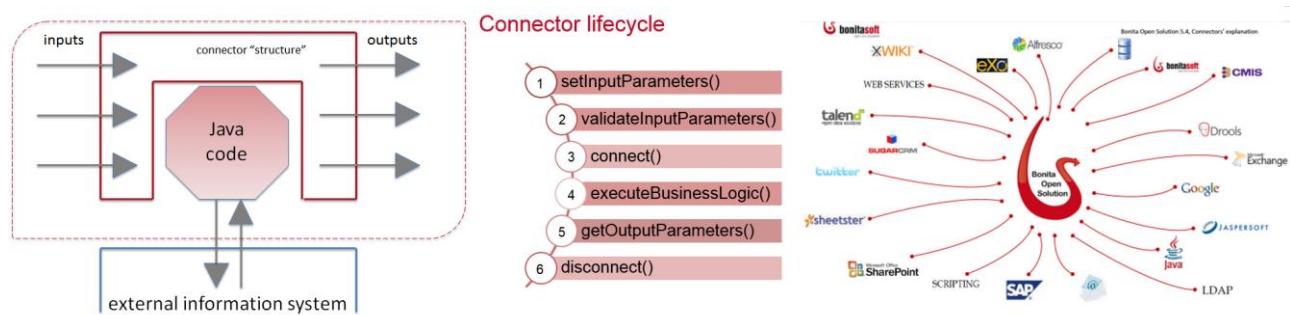
- D'un studio de développement (Bonita BPM Studio) permettant de développer des modèles BPMN. Il permet de modéliser graphiquement des processus puis de connecter ces processus aux composants du système d'information (messagerie, base de données, services tiers etc.) afin de générer une application métier autonome, codée en Java et accessible depuis une interface web. Le Studio permet également de personnaliser les formulaires tant dans leur ergonomie que dans leur apparence graphique.

- Un moteur BPM (Bonita BPM Engine) utilisable dans des applications autonomes et pouvant gérer facilement des processus complexes avec un haut volume de transactions. Il s'intègre dans tous les environnements, et est distribué sous licence LGPL.
- Un portail web (Bonita BPM Portal) s'exécutant via les serveurs d'application Jboss ou Tomcat. Il permet d'exécuter les processus, gérer les utilisateurs et leurs droits ainsi que de suivre le déroulement des processus via des indicateurs statistiques.

De plus la société BonitaSoft a créé un véritable écosystème autour de sa suite BPM directement intégré à l'écran de démarrage du logiciel. Il est ainsi possible d'accéder à de la documentation, des formations (payantes), des podcasts, des exemples et une communauté de partage.

La notion de connecteur dans Bonita :

Les connecteurs servent à relier de manière simplifiée le processus aux systèmes d'information externe. Les connecteurs représentent des actions exécutés par le système.



Bonita intègre des connecteurs préconçus qu'il suffit de paramétriser. En pratique la plupart des connecteurs nécessitent des connaissances en communication des systèmes d'information pour être intégrer efficacement. Bonita fournit également un outil pour concevoir ses propres connecteurs partageables ensuite avec la communauté.

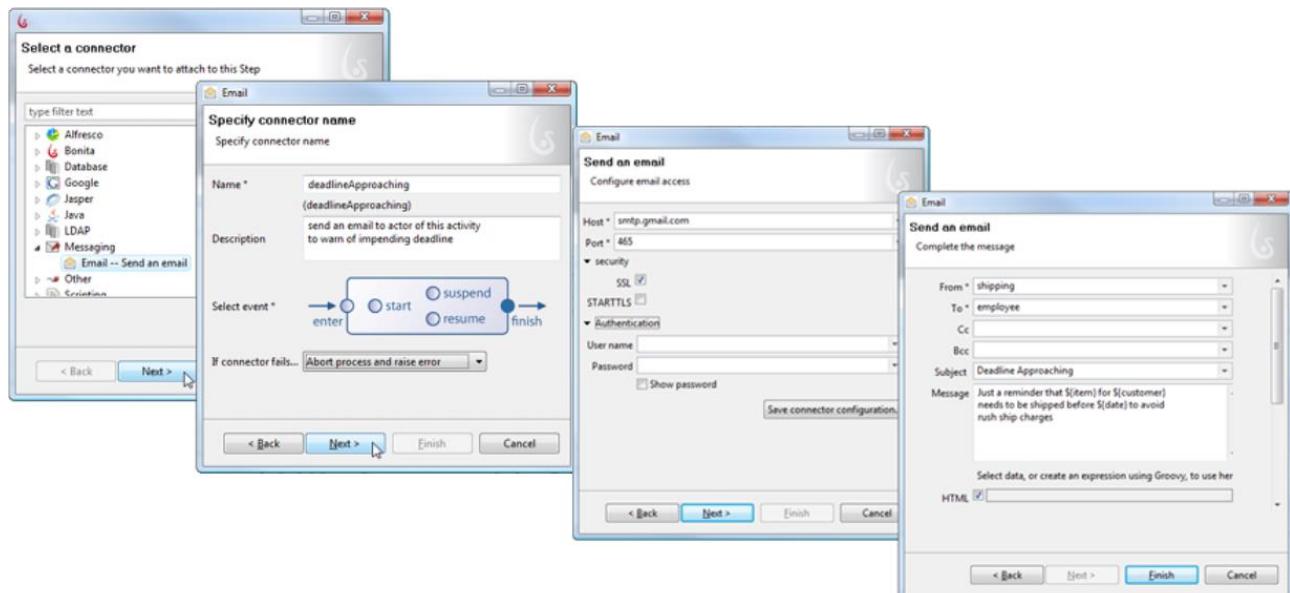


Figure 29: Connecteur générique à un service de messagerie

CONCLUSION

À l'issu de ce chapitre, l'usage de BPMN 2.0 dans une perspective éducative nous paraît posséder les avantages suivants :

- **BPMN est un Standard:** il n'est pas la propriété d'une entreprise donnée, mais est défini par un consortium (OMG) influant et apte à diffuser et faire évoluer la spécification. Ce statut laisse entrevoir des possibilités d'interopérabilité entre outils et de réutilisation des modèles intéressantes pour l'éducation. De plus la pérennité d'un langage est essentielle pour justifier les efforts entrepris pour son adoption.
- **BPMN est une notation abstraite:** De ce point de vue elle semble utilisable dans une perspective éducative.
- **BPMN est soutenu et utilisé :** BPMN est activement développé et dispose d'une industrie puissante apte à développer des outils permettant son utilisation. Des formations publiques ou privées sont disponibles et une communauté importante de professionnel peut soutenir son utilisation effective.
- **BPMN définit une sémantique d'exécution :** Ce qui offre des perspectives intéressantes d'implémentation rationalisant l'effort de modélisation. En éducation, les modèles BPMN pourraient ainsi faire le lien entre les intentions pédagogiques et les processus de conception des EIAH. De plus une certaine approche de conception impliquant peu de programmation semble se diffuser et laisse entrevoir un développement simplifié pour un public moins technique.
- **BPMN concerne différents corps de métier.** L'utilisation d'un langage trop spécialisé ne nous semble pas viable pour une collaboration multi-acteurs efficace. Le développement d'un dispositif de formation implique de nombreux acteurs issus de divers horizons professionnels. Pour un ingénieur pédagogique étudier un langage connu des administrations, de sa hiérarchie ou des équipes IT est un avantage non négligeable.
- **BPMN s'accompagne d'outils puissants :** BPMN est au cœur d'une approche de conception orientée services. Les BPMS offrent une plateforme de développement apte au développement d'applications flexibles, exécutables sur différentes plateformes et interconnectable avec tous services externes. S'il n'existe pas à l'heure actuelle de BPMS adapté à un public non-technique, les BPMS représentent cependant une infrastructure intéressante dans les projets de formation importants impliquant une équipe de développement.

À contrario nous pouvons soulever des interrogations liées au manque de visibilité du langage en contexte pédagogique. La recherche commence à explorer le sujet et les travaux sont encore rares. L'adéquation du langage avec les pratiques pédagogiques reste à investiguer. En outre, aucun outil spécialisé pour un usage en contexte éducatif n'existe et aucune « guideline » ne peut véritablement soutenir le praticien à l'heure actuelle. Par conséquent, il n'existe pas non plus de communauté de pratique permettant le partage de connaissances sur le sujet ou la mise à disposition de scénarios BPMN. Cette situation peut bien entendu évoluer à l'avenir et ce mémoire apporte une contribution à cet enjeu de visibilité.

Au chapitre suivant, nous aborderons BPMN en contexte éducatif et proposerons, dans une visée d'implémentation, de modéliser des scénarios avec une palette d'éléments graphiques BPMN. Nous déployerons une plateforme de gestion de workflow grâce aux outils de développement d'un BPMS et testerons l'exécution des scénarios pédagogiques sur cette dernière.

CHAPITRE V. MODÉLISATION DE SCÉNARIO PÉDAGOGIQUE AVEC BPMN

« With the right modelling tools that have intuitive graphical flow diagrams, graphical modelling of pedagogies can provide a platform for strategic pedagogical planning that describes the complexity of learning processes through structured course materials. Course designers can design pedagogical templates that can: be used and reused; be collaboratively developed for teaching and learning processes; and, capture various pedagogical scenarios (i.e., interactions with learning resources by learners) and the intricacies associated with learning activities. »

[Adesina, 2013, p. 85]

Dans ce chapitre nous étudierons la possibilité d'utiliser BPMN comme un langage de modélisation pédagogique. L'abstraction du langage, sa volonté d'être un langage compréhensible pour des personnes non-techniques et son usage ayant fait ses preuves dans d'autres domaines d'ingénierie offrent des perspectives intéressantes. De plus la définition récente d'une sémantique d'exécution pourrait en faire un candidat de choix pour faciliter les processus d'implémentation de dispositifs instrumentés. Cependant BPMN est un langage très général de conception et il convient de définir les cadres de son utilisation pour un usage en éducation. Nous proposerons une palette d'éléments concrets pour son utilisation dans un cadre éducatif et introduirons des bases pour aborder la modélisation. Nous déployerons une plateforme de gestion de workflows et explorerons les potentialités d'exécution de nos scénarios.

INTRODUCTION

Si BPMN est mentionné dans la littérature comme une alternative aux langages de modélisation pédagogique développés spécifiquement pour le domaine éducatif (tel que IMS Learning Design), il ne semble pas y avoir beaucoup d'investigations approfondies quant à son potentiel en éducation. À l'heure actuelle deux projets semblent récemment s'intéresser à l'utilisation du langage BPMN ou à l'architecture BPM dans le cadre de la conception d'environnement d'apprentissage.

- Le projet de thèse de Adesina, A. dirigé par Molloy, D. (2013) présente un environnement d'apprentissage virtuel de processus (VLPE) basé sur le framework Business Process Management (BPM), où les concepteurs de cours utilisent une palette très réduite d'élément BPMN. L'idée est de proposer une application « standalone » permettant de modéliser des parcours d'apprentissage via 7 éléments BPMN. Ces parcours sont ensuite implémentés via un transfert vers le langage de d'exécution BPEL et géré par une application BPM qui grâce à différentes API java est capable d'exécuter et traiter les pédagogies modélisées. Le système n'utilise donc pas les récentes fonctionnalités de BPMN 2.0. BPMN n'est pas utilisé comme un standard pour la modélisation de scénarios interopérables et exécutables.
- Le projet MEDUSY (Multi-Purpose EDUcation SYstem) de l'université Masaryk (république tchèque) en partenariat avec l'entreprise Américaine RedHat développe actuellement une solution pour la conception d'environnements d'apprentissage riches basés sur les processus pour la gestion de l'apprentissage. Ce projet open-source a pour objectifs : de fournir des informations sur l'approche processus et de montrer comment il peut être mis à profit dans le contexte du e-learning ; d'identifier des modèles d'apprentissage générique ; de fournir un environnement pour la modélisation, l'exécution et le suivi des processus d'éducation et enfin d'offrir une plate-forme e-learning flexible et modulaire permettant de brancher des services externes. Dans ce contexte, BPMN 2.0 est actuellement jugé « la plus

intéressante quant à son potentiel». Il existe encore peu d'information en anglais sur ce projet tchèque. Cependant Jiří Kolář et Tomáš Pitner (2011) semble être les premiers auteurs à utiliser BPMN 2.0 en tant que langage de modélisation pédagogique.

Dans ce mémoire nous partageons la vision du projet MEDUSY et contribuons à l'étude de BPMN 2.0 sous un paradigme éducatif avec une approche moins techno-centré.

1. BPMN SOUS UN PARADIGME ÉDUCATIF

De nombreux travaux de recherche récents proposent de représenter les scénarios pédagogiques sous une forme graphique (Paquette 2002 ; Vantroys & Peter 2005 ; Lafocade 2005 et 2006 ; Martel & al. 2006 ; Botturi & Stubbs 2008 ;). Ces langages permettent de « *supporter la créativité des concepteurs* » et « *rendre possible la discussion au sein d'une équipe de conception [...] pour assurer la nécessaire médiation entre les processus pédagogiques exprimés et les processus que sont capables de mettre en œuvre les infrastructures informatiques actuelles* » (Nodenot, 2008, p.94). Pour autant il n'existe pas de réel standard aujourd'hui en la matière. Comme le constate Karampiperis & Sampson (2007) :

« En dépit de la large adoption de la spécification IMS-LD, il manque toujours un langage commun pour représenter graphiquement les scénarios d'apprentissage. Cela est dû au fait que, bien que la spécification IMS LD fournit les moyens pour la représentation de flux d'apprentissage technique, il ne donne pas d'indications sur la façon dont ces flux pourraient être représentés d'une manière humaine compréhensible. En conséquence, il existe plusieurs outils de création pour la conception d'activités d'apprentissage permettant d'exporter dans un format conforme à IMS Learning Design, mais ces outils utilisent des représentations différentes du processus d'apprentissage. D'autre part, il existe des normes pour la création de représentations graphiques humainement compréhensible des processus (à savoir le Business Process Modelling Notation standard), qui pourraient être utiles pour procurer une représentation communes des scénarios d'apprentissage. »

BPMN est un standard international en pleine expansion et activement développé. Il est aujourd'hui assez largement adopté et est enseigné dans des formations publiques ou privées. Étant conçu pour être utilisé par un personnel non technique (manager, analyste métier) afin de mieux collaborer avec les équipes de développement, il peut être intéressant d'investiguer son possible usage dans le domaine éducatif.

1.1 BPMN en tant que Langage graphique de modélisation pédagogique.

Selon les 5 propriétés présentées par Martel, Lejeune, Ferraris & Vignollet (2007, p.513) caractérisant un langage de modélisation pédagogique (voir chapitre 3), BPMN peut être utilisé comme un LMP. C'est une notation abstraite et métaphoriquement neutre. Elle permet d'exprimer les différentes relations possibles entre des activités (échanges de ressources et informations, synchronies/asynchronies des activités etc.). De plus elle permet de situer les actions et l'expression des acteurs et de leurs rôles respectifs aux seins d'une situation d'apprentissage par l'intermédiaire des éléments *bassins* et *coulloirs* encadrant les activités de chacun dans le processus. Nous pouvons comparer BPMN à d'autres langages de modélisation selon les critères de Botturi & al. (2006) et Nodenot & al. (2008) :

Botturi & al. (2006) :

Ces auteurs proposent une classification des langages visuels de modélisation pédagogique selon 5 critères : le niveau de *stratification* (un ou plusieurs niveaux d'abstraction), le niveau de *formalisation* (d'informel à formel), son niveau d'*élaboration* (conceptuel, spécification, implémentation), le nombre de *perspectives* (unique ou multiple) qu'il propose pour analyser une situation pédagogique et le *système de notation* qu'il offre aux concepteurs (textuel ou visuel). Ci-dessous nous ajoutons BPMN 2.0 à une classification effectuée par Nodenot & al. (2008) selon ces critères.

	Niveau de stratification	Niveau de formalisation	Niveau d'élaboration	Nombre de perspectives	Notation
E ² 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
BPMN 2.0	plusieurs couches	formel	spécification / implémentation	une	visuel

Tableau 1: Comparaison de BPMN 2.0 à d'autre LMP selon critères de classification de Botturi & al. (2006).

Nodenot & al. (2008) :

L'auteur propose une classification complémentaire à la précédente composée de 7 critères.

1. La représentation des rôles et responsabilités que prennent les acteurs dans les situations d'apprentissage décrites.
2. La représentation des modalités d'apprentissage prescrites. Ces modalités 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.
3. La représentation des connaissances sous-jacentes aux activités d'apprentissage prescrites.
4. La représentation précise des buts d'apprentissage qui, pour certains langages visuels, doit rester conforme à des modèles théoriques tels que décrits dans Gagné, Briggs et Wager (2005) ou Merrill (1994).
5. 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.
6. La représentation des collaborations synchrones/asynchrones entre acteurs participant à une situation d'apprentissage.
7. La représentation de liens avec l'infrastructure.

Le tableau 8 présente la classification de différents langages analysés par l'auteur, auquel nous ajoutons une comparaison avec BPMN 2.0.

	E ² ML	CPM	coUML	MOT+	PoEML	ASK-LDT	COLLAGE	BPMN 2.0
Rôles et responsabilités		+	+		+	+	+	+
Modalités d'apprentissage	+		+			+		+
Domaine, connaissances	+	+	+	+				
Objectifs et buts d'apprentissage	+	+		+				
Structure du cours	+			+	+	+		+
Collaborations entre acteurs	+	+	+			+	+	+
Liens avec l'infrastructure		+	+	+	+	+		+

Figure 30: Comparaison de BPMN 2.0 à d'autre LMP selon critères de classification de Nodenot (2008).

Ce tableau rapproche BPMN des langages CoUML et ASK-LDT. Tout comme ces deux langages BPMN permet de définir les activités d'un scénario pédagogique en axant la représentation sur les rôles et responsabilités de chacun, leur collaboration et leurs interactions avec le système.

1.2 Cycle de vie d'un scénario pédagogique et approche BPM.

Au point 1.1 nous présentions le cycle de vie des processus métier conçu autour d'une phase d'analyse /modélisation (BPA), une phase d'implémentation technique (BPI) et une phase de contrôle/évaluation (BAM). Ces phases sont très similaires au cycle de vie de la conception d'un scénario pédagogique. Selon Vantroys (2003, p.11-12) le processus de conception d'un module d'enseignement comprend 6 phases.

1. *Expression initiales des besoins et des modèles.* Cette phase correspond à une description informelle par l'enseignant des ressources, des activités et de la pédagogie adoptée pour le module.
2. *Analyse et conception.* Un ingénieur pédagogique formalise le scénario en collaboration avec l'enseignant.
3. *Implémentation.* Réalisation des ressources techniques et métiers.
4. *Déploiement.* Processus de diffusion des éléments développés dans les phases précédentes. Ce déploiement se fait classiquement par l'intermédiaire d'une plateforme type LMS.
5. *Test.* Vérification du comportement du dispositif d'un point de vue technique et fonctionnel.
6. *Évaluation.* Évaluation du comportement des apprenants et enseignants en interaction avec le dispositif. Définition des ajustements nécessaires pour une éventuelle remise en pratique du dispositif.

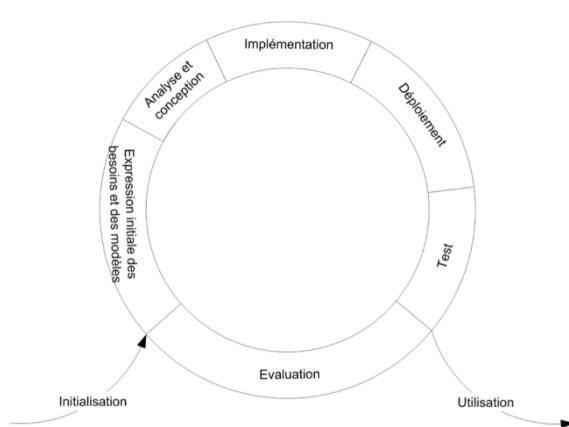


Figure 31: cycle de vie d'un module d'enseignement selon Vantroys (2003).

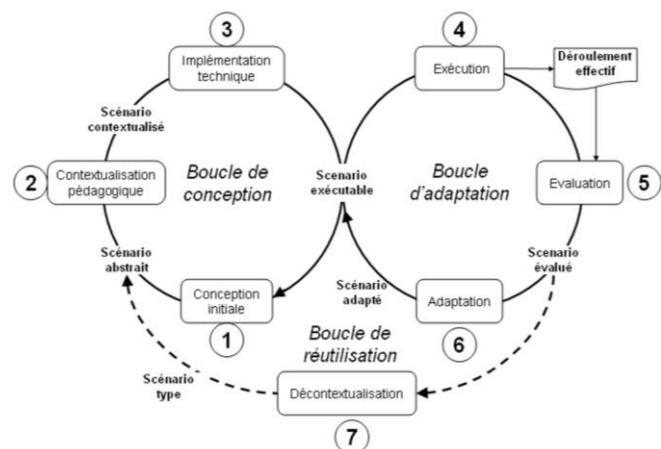


Figure 32: Cycle de vie d'un scénario pédagogique selon Pernin (2007).

Pernin (2007) quant à lui, propose un modèle en 7 étapes (très similaire à celle de Vantroys) auquel il inclut l'état d'un scénario pédagogique à chacune des phases. Pernin distingue les étapes suivantes : la conception initiale, la contextualisation pédagogique, l'implémentation technique, l'exploitation, l'évaluation, l'adaptation et la réutilisation. Ce cycle de vie définit la façon dont les orchestrations sont conçues, mises en œuvre, ajustées ou réutilisées. Le modèle insiste sur le processus de réingénierie des scénarios pédagogiques à travers le formalisme d'un LMP.

Le développement d'un scénario pédagogique avec BPMN se prête parfaitement selon nous à la mise en application des différentes étapes présentées ci-dessus. BPMN est une notation formelle, abstraite et standardisée, ce qui favorise la décontextualisation, le partage et l'interopérabilité. Les suites BPMS conçues pour implémenter des processus métier selon un cycle similaire pourrait avantageusement être employé pour la conception de scénarios pédagogiques. Notamment, concernant les phases de déroulement et d'évaluation des scénarios où les BPMS procurent des mécanismes intéressant d'analyse de données en temps réel (voir point suivant).

1.3 Analyse des traces de processus et analyse de données de formation.

« Un des effets secondaires de l'Internet, des ordinateurs, des appareils mobiles et des systèmes de gestion de d'apprentissage en ligne (LMS) est le passage d'information éphémère à la capture de données explicites. L'écoute d'un exposé en classe ou en lisant un livre laisse un nombre de traces limité. Une conversation de couloir s'évapore essentiellement dès qu'elle est terminée. Cependant, chaque clic, chaque Tweet, chaque mise à jour de son statut Facebook, chaque interaction sociale, et chaque page lue en ligne peuvent laisser une empreinte numérique. En outre, l'apprentissage en ligne, les dossiers numériques des étudiants, les cartes étudiantes, les capteurs et appareils mobiles captent maintenant des traces riches en informations, captent les flux d'activité. »

[Long & Siemens, 2011, p.2]

L'exploitation des données en circulation dans les systèmes informatiques a très vite été exploitée dans les champs économiques et politiques. Un champ de recherche, que l'on qualifie d'informatique décisionnelle (Business intelligence), s'est développé autour de cette question afin d'étudier les moyens, outils et méthodes qui permettent de collecter, consolider, modéliser, résumer et restituer les données en vue d'offrir une aide à la décision. Dans le champ de l'éducation, l'exploitation des données de formation disponible dans les environnements numérique d'apprentissage a été plus tardive mais semble aujourd'hui se fédérer sous la terminologie de « Learning Analytics ». Selon la première International Conference on Learning Analytics and Knowledge (2011), nous pouvons la définir comme « *la mesure, la collecte, l'analyse et l'élaboration de rapports sur les données concernant les apprenants et leurs contextes, à des fins de compréhension ou d'optimisation de l'apprentissage et des environnements dans lesquels il se déroule* ». Pour Long et Siemens (2011), le champ des « Learning Analytics » est très spécifique et se concentre sur les processus d'apprentissage. Il est ainsi important de ne pas confondre avec ce qu'ils nomment « Academic Analytics », relevant simplement de l'informatique décisionnelle dans l'enseignement académique à un niveau institutionnel, régional, national ou international.

Les Learning Analytics rencontrent aujourd'hui un intérêt croissant en technologies éducatives. Dans la formation entièrement à distance on espère mieux comprendre l'activité « invisible » des étudiants. À grande échelle, notamment avec le récent engouement pour les MOOC, ils nourrissent l'espoir d'une construction de modèles prédictifs d'apprentissage basées sur les « big data ». Dans une perspective de réingénierie, ils ont vocation à fournir des données concrètes pour évaluer et améliorer les scénarios pédagogiques ou les environnements numériques. Enfin de manière générale, ils doivent permettre aux équipes enseignantes de mieux piloter leur formation en fournissant des données quotidienne sur l'activité des apprenants au sein du dispositif. On retrouve ainsi l'analyse de données a priori (prédition par rapport au modèle), pendant (pilotage, gestion) ou a posteriori (Évaluation, optimisation). L'analyse peut se focaliser sur le collectif (Ex : les difficultés les plus fréquemment rencontrées par les apprenants) ou sur un seul apprenant (Ex : Compréhension du profil étudiant à travers les traces numériques).

Comme nous avons pu l'évoquer au chapitre 4, l'informatique décisionnelle fait partie intégrante de l'approche BPM. La BAM (Business Activity Monitoring) consiste à combiner les informations puisées dans plusieurs systèmes pour constituer des tableaux de bord d'indicateurs métiers destinés aux responsables opérationnels. Tous les BPMS propose un moyen d'obtenir des rapports d'activité. La méthode consiste généralement à constituer des indicateurs clés de performance : les KPI (Key Performance Indicator). La définition d'un KPI intervient durant la phase de modélisation et consiste à mesurer une variable ou opérer un calcul sur un ensemble de variables. Le KPI est ensuite attaché au processus. À chaque fois que le processus sera lancé, les données assignées au KPI seront collectées et sauvegées dans une base de données spécifiée. L'utilisateur final peut ensuite générer des rapports via le portail d'administration. Il est en général possible d'utiliser en parallèle des outils de génération de rapports puissants conçus pour fonctionner sur les serveurs d'applications

Java tels que JasperReports ou Birt Reporting. Ces outils spécialisés permettent d'utiliser des modèles de rapport avec des visualisations dynamiques avancées et de multiples formats d'export.

Pour notre approche de modélisation des scénarios pédagogiques avec BPMN, la perspective d'une utilisation des KPI est intéressante. Elle nécessite cependant de définir à l'avance les points sur lesquels centrer le regard et se prête mal à l'analyse exploratoire a posteriori des données de formations. De plus il faut garder à l'esprit que le dispositif ne capture qu'une partie de l'activité de l'apprenant (celle où il est en interaction avec le système). L'approche par les KPI se focalise sur l'exécution des tâches, dans une perspective de gestion, d'évaluation et d'optimisation de ces derniers. Comme son nom l'indique le but premier est la performance. Pour autant ils fournissent un mécanisme efficace et puissant de gestion des données de formation en temps réel pouvant permettre, à titre d'exemple :

- De surveiller le bon déroulement des activités d'apprentissages avec un haut niveau de granularité (tâches ouvertes, réalisées, en retard, en échec, non ouvertes etc.) ;
- De repérer rapidement, via le déclenchement d'alerte, un besoin de soutien à l'apprenant lorsque certaines valeurs critiques sont atteintes ;
- D'adapter de manière automatisée le parcours d'un apprenant en fonction de sa performance
- De situer un apprenant par rapport à l'activité du reste du groupe ;
- D'identifier les tâches les plus problématiques pour les apprenants, (nombre de régulations nécessaires, temps de résolutions etc.) ;
- De corrélérer les données obtenues durant l'exécution d'un processus aux mêmes données obtenues lors des exécutions précédentes avec d'autres cohortes d'apprenants.

2. PROPOSITION D'UTILISATION

Dans cette section nous proposons des principes d'utilisation de BPMN dans le cadre des technologies éducatives. L'objectif étant de donner des points de repère pour aborder une modélisation faisant sens d'un point de vue des technologie éducative mais respectant le standard et l'approche BPM afin de pouvoir être compris par les autres acteurs impliqués dans le processus de conception. Notre proposition reste fidèle à notre cadre conceptuel définissant la scénarisation, sous l'angle d'un workflow centré sur la notion d'activité. Afin d'aborder la modélisation nous commencerons par définir une palette d'éléments graphiques en les organisant et définissant sous l'angle de leur emploi possible en technologies éducatives. Nous présenterons ensuite un cadre d'utilisation pour modéliser avec BPMN selon notre approche. Enfin nous proposerons de les exécuter sur « E-Flow » une plateforme que nous avons déployé à partir d'un BPMS.

2.1 Palette d'éléments.

Nous avons organisés des éléments BPMN que nous considérons comme nécessaires/utiles pour la modélisation de workflow pédagogique. Nous avons organisé les éléments en 6 catégories, tentant de faire correspondre la norme BPMN à une perspective d'ingénierie pédagogique. Pour être plus clair, l'objectif est de réaliser un compromis en effectuant un regroupement ne trahissant pas la nature des éléments graphiques (leur définition et leur emploi selon la norme OMG) mais faisant sens pour la conception d'un scénario pédagogique (ex : la notion de temps dans une activité). Ainsi, nous obtenons les 6 catégories suivantes :

1. *Actions* : Comprend les éléments graphiques permettant de représenter les actions/tâches effectuées par les acteurs et services informatiques au cours d'un workflow.
2. *Déroulement* : Comprend les éléments permettant de représenter les parcours (soit l'enchaînement des actions et la manière dont ces dernières se distribuent entre les acteurs du dispositif).
3. *Temporalité* : Démarrage, arrêt, pause des différentes activités/actions en fonction du temps ou du déroulement d'autres activités/actions.

4. *Ressources et productions*: Permet de formaliser les intrants et extrants à une action/activité (ex : La synthèse rédigée par un étudiant).
5. *Décomposition/regroupement* : Permet l'organisation visuelle des diagrammes et la structuration en différents workflow autonome et inter-opérant.
6. *Artefacts*: Comprend les commentaires (tel que définis par la norme BPMN) et tous éléments additionnels ajoutés par le modélisateur pour la compréhension des diagrammes.

I. Actions		
	Action humaine	Permet de modéliser une action effectuée par un apprenant ou un autre acteur du dispositif <u>en interaction</u> avec un service informatique.
	Action effectuée par un service	Action effectuée par un service informatique sans intervention d'un acteur du dispositif.
	Action manuelle	Action effectuée par un acteur <u>sans interaction</u> avec un service informatique.
	Action de communication	Cette action permet de représenter une tâche spécifique de communication.
	Action abstraite	Permet une modélisation neutre sans entrer dans les détails.
II. Déroulement		
	Flux d'action	Permet de représenter le parcours d'apprentissage à travers le déroulement de l'activité (connexion des différentes actions entre elles).
	Flux de communication	Permet de représenter les communications entre différents processus (bassin) distincts.
	Branchement exclusif	Division du parcours d'apprentissage en différents chemins. Un seul des chemins peut être emprunté. Des conditions régissent le choix de parcours.
	Branchement parallèle	Division du parcours en différents chemins. Chacun des chemins comportent des actions pouvant être effectuées de manière parallèle ou séquentielle.
	Répétition, duplication.	3 éléments graphiques pouvant être ajoutés dans la partie inférieur du symbole « appel d'activité » et de tous les symboles de la catégorie « Actions ». Représentent respectivement, une action répétée en boucle, dupliquée et effectuée en parallèle, dupliquée et effectuée de manière séquentielle.
	Couloir de rôle	Les rôles ou acteurs sont modélisés par des couloirs compartimentant le scénario et permettant de spécifier quelle action est effectuée par quel rôle/acteur.
III. Temporalité		
	Départ et Arrêt	Permet de spécifier le démarrage (trait fin) et l'arrêt (trait foncé) d'un workflow. Un workflow a toujours un début et une fin.
	Arrêt et départ avec signal	Surtout utilisés dans une perspective d'implémentation. La fin d'un workflow envoie un signal pouvant être attendu pour le démarrage d'un second workflow.
	Signal intermédiaire (Synchronie)	Fonctionnement similaire à la ligne précédente, mais au cours du workflow. Permet notamment de synchroniser des workflows. Cet évènement peut être utilisé en bordure d'une action pour envoyer un signal durant une tâche.
	Arrêt et départ avec données	Similaire aux éléments « signal », cependant des données accompagnent ici le signal. Un message <u>n'est pas un e-mail</u> , mais des données (constantes, variables, paramètres etc.) transmises au processus récepteur.
	Donnée intermédiaire	Similaire au signal intermédiaire mais contient des données.
	Départ programmé	Le workflow se lance à une date spécifiée ou selon un cycle donné (ex : tous les matins à 9h).

	Minuterie (interruatrice ou non)	Permet de spécifier à quelle date ou au bout de combien de temps le workflow se poursuit, s'arrête etc. Cet évènement peut être utilisé en bordure d'une action pour spécifier un chemin alternatif si l'action n'est pas réalisée au bout de X temps (en interrompant ou non la tâche en cours).
IV. Ressources et productions		
	Ressource	Ressource associée, utilisée ou nécessaire à la réalisation d'une action/activité. Trois barres verticales signifiant de multiples ressources.
	Production	Produit découlant de la réalisation d'une action/activité. Trois barres verticales signifiant de multiples productions.
	Donnée transitant	Information parcourant le workflow, « consommé » au cours de l'activité. Généralement le produit d'une action servant de ressource pour une autre.
V. Décomposition et regroupement		
	Regroupement	Permet d'encadrer d'autres éléments afin de les regrouper visuellement. Ces groupes peuvent être nommés et commentés et permettent de structurer un workflow complexe en unité visuelle.
	Appel d'activité	Permet d'encapsuler le workflow d'une activité autonome. Cet élément peut être représenté ouvert (révélant l'activité qu'il contient), ou fermé (contenu non-visible). Certains formats d'export du diagramme permettent l'ouverture/fermeture dynamique de l'activité par un clic.
	Appel d'activité Ad Hoc	Similaire à l'élément précédent mais contient aucun workflow. Simplement une liste d'actions, ressources, artefacts dont le déroulement n'est pas spécifié.
	Lien	Permet de connecter deux sections d'un même workflow sans les lier par un flux d'action. Éléments permettant la plupart du temps de simplifier la lecture d'un diagramme. Plusieurs liens-sources peuvent viser un lien-cible mais pas l'inverse.
VI. Artefacts		
	Commentaire	Permet d'introduire tout type de commentaires de manière standardisée dans un diagramme BPMN.
	Élément additionnel	Tout élément graphique que le modélisateur juge nécessaire d'ajouter pour permettre une meilleure compréhension du diagramme ou l'ajout d'informations importantes non représentables par les éléments précédents. (Tel que permis dans la spécification BPMN). À utiliser avec parcimonie car potentiellement non compréhensible par des acteurs non-initiés à sa signification.

Exemple illustratif 1 - (Gestion temporelle)

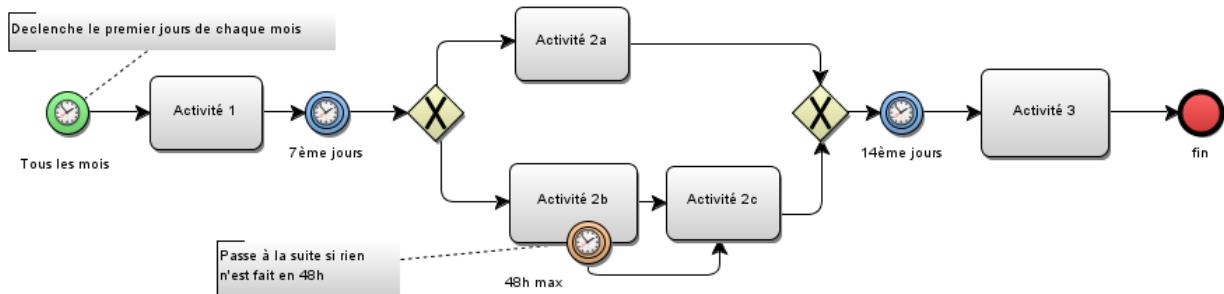


Figure 33: Exemple illustratif - Gestion temporelle en BPMN

Exemple illustratif 2 - (Synchronie)

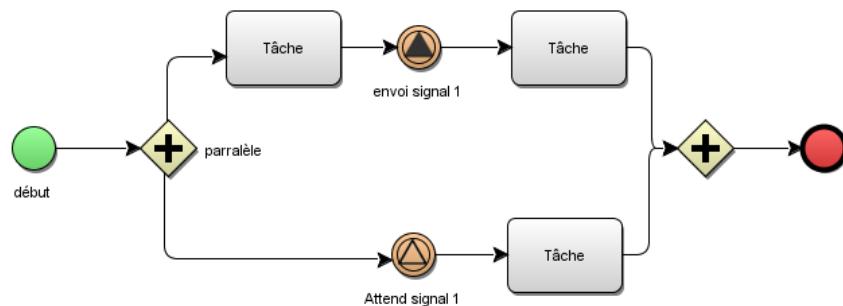


Figure 34 : Exemple illustratif 2 : Synchronie en BPMN

Exemple illustratif 3 – (Ressources, données, produits)

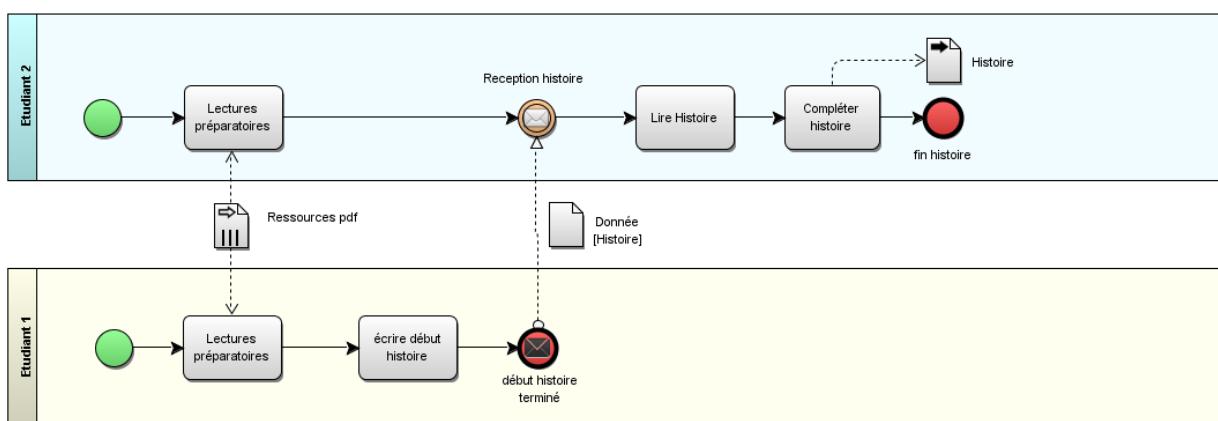


Figure 35: Exemple illustratif 3 : Ressources, données, produits en BPMN

2.2 Cadre de conception.

La modélisation est toujours un acte difficile. Pour l'aborder il convient de se doter d'un cadre.

Perspectives/ finalités :

Lorsque l'on aborde la modélisation avec BPMN il est nécessaire selon nous de se positionner dès le départ selon une double perspective. La perspective d'implémentation et la perspective de communication. Ces deux perspectives sont complémentaires mais ne visent pas les mêmes objectifs et ne s'abordent pas de la même façon.

- La perspective communicationnelle permet une modélisation plus libre et centrée sur les intentions pédagogiques des concepteurs. Elle permet de structurer et formaliser des idées. Elle soutient l'activité créatrice des concepteurs. Une bonne modélisation est un processus comprenant de multiples rétroactions entre les différents acteurs impliqués dans sa conception afin d'arriver à une solution jugée optimale et unanimement partagée. Le diagramme ainsi obtenu pourra servir de documentation au dispositif et de point de départ pour discuter avec une équipe de développement. Dans une perspective à plus long terme les diagrammes peuvent être corrigés et améliorés au fil des sessions de formation et des évaluations. La réingénierie des scénarios permet d'optimiser le dispositif à chaque itération et de rationaliser l'effort de conception.
- La perspective d'implémentation se concentre sur une modélisation plus opérationnelle, c'est-à-dire plus précise sur la mécanique du workflow mais parfois plus incomplète en ce qui concerne l'aspect formatif. Elle se conçoit en général comme une modification/transformation des diagrammes issus d'une perspective communicationnelle. Durant son élaboration, les rétroactions se situent plutôt du côté acteur-

système afin de tester le comportement correct du dispositif implémenté au regard des objectifs visés par la modélisation. Lors de la modélisation le concepteur doit constamment imaginer la manière dont le système va traduire le scénario.

Décomposition/Granularité :

À la double perspective précédente, il est nécessaire d'associer un positionnement sur la granularité de la modélisation avec BPMN. Afin d'aborder cette dernière nous proposons d'utiliser les concepts de la TA (Chapitre 2). Notre modélisation avec BPMN se base sur la notion d'activité au sens de cette dernière en tant que succession d'actions et d'opérations socialement distribuées en interaction avec un environnement et orientées vers un objet. L'objet d'une activité est la production matérielle ou immatérielle d'un résultat perceptible témoignant de l'atteinte de celui-ci. Une activité est ainsi à concevoir comme une unité de sens nous permettant d'aborder la structuration d'un scénario lors de la modélisation et n'a donc pas nécessairement de correspondance avec la conception d'activité pédagogique telle qu'elle peut se concevoir en sciences de l'éducation. Dans notre approche une activité peut être pédagogique ou non pédagogique (ex : activité de mise en route d'un cours par l'équipe enseignante).

Notre proposition de développement avec BPMN s'articule autour d'un double niveau de granularité:

1. *Niveau 1 : L'orchestration des activités.* L'organisation des activités entre elles, les liens temporels et hiérarchiques qu'elles entretiennent de telle sorte à atteindre les objectifs d'apprentissage. Cette perspective utilise abondamment l'élément « appel d'activité » défini dans le tableau d'éléments de la section précédente pour représenter les activités à un niveau macroscopique.
2. *Niveau 2 : Le workflow d'une activité élémentaire.* Le déroulement d'une activité décrit les processus interactionnels des acteurs dans un dispositif composé d'humains, de ressources et d'artefacts informatiques en vue de l'atteinte d'un objectif spécifié.

Ces deux niveaux représentent la base de notre approche. Il est en outre possible d'ajouter un troisième niveau de manière plus micro – « niveau 3 » - pour décrire une petite chaîne critique d'opérations nécessaires à la réalisation d'une tâche de l'activité. Quant à un niveau plus macro – « niveau 0 » - il reste envisageable mais aura souvent peu de pertinence (la structuration à un tel niveau dépend généralement peu d'un workflow).

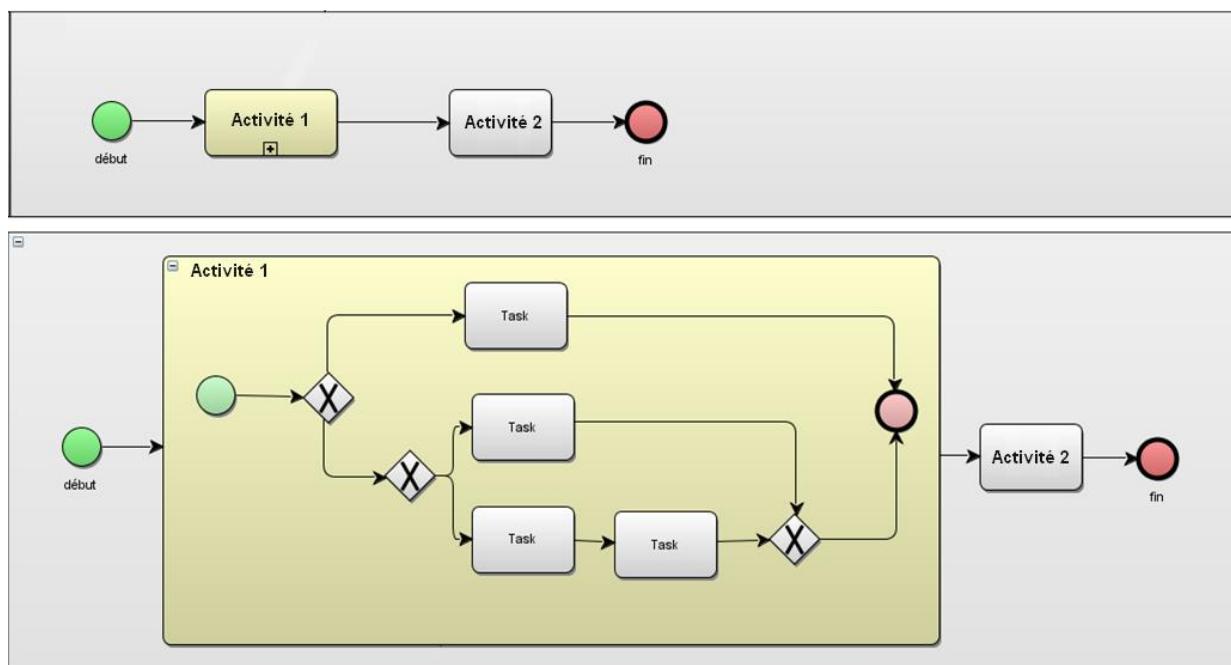


Figure 36: Principe de décomposition. Niveau 1 et 2.

En croisant ces deux niveaux de granularité selon les deux perspectives précédentes nous obtenons un tableau définissant 4 dimensions pour aborder la modélisation avec BPMN.

	COMMUNICATION	IMPLEMENTATION
NIVEAU 1	Conception avec rétroactions acteurs-acteurs	Conception avec rétroactions acteurs-système
NIVEAU 2	2 Description du workflow d'une activité. « Qui fait quoi à quel moment dans quelles conditions et avec quels outils ». Le diagramme conçu doit pouvoir se transformer en diagramme exécutable moyennant des modifications opérationnelles.	3 Centration sur l'automatisation de l'orchestration. La manière dont le système lancera toutes ou certaines activités en fonction de paramètres définis (Temps, réception d'un signal, résultat d'un apprenant à une activité etc.).
		4 Centration sur l'automatisation d'une des activités du scénario afin de créer une petite application autonome. L'activité pourra être démarrée manuellement par un utilisateur ou automatiquement par le système suivant le workflow de niveau 1.

Tableau 2 : Perspective et niveau de granularité lors de la modélisation de workflow BPMN

Initialisation/Instanciation :

Dans le cas d'une perspective d'implémentation, un autre aspect doit être pris en compte lors de la modélisation avec BPMN et concerne l'initialisation du workflow et l'instanciation des différentes tâches. La modélisation n'est pas la même lorsque l'on aborde un scénario exécuté par un enseignant et un scénario exécuté au niveau de chaque apprenant. De plus ces deux manières différentes de procéder sont fortement liées au niveau de granularité du scénario.

	NIVEAU 1	NIVEAU 2
ENSEIGNANT	1 Scénario global initialisé par l'enseignant. Workflow lancé une seule fois. Toutes les activités ou tâches des apprenants doivent être multi-instanciées en parallèle et attribuées à chaque apprenant. Permet l'orchestration globale mais est assez complexe à implémenter.	2 Concerne une activité pouvant être lancée de multiple fois par un enseignant, par exemple une activité de régulation avec un étudiant. Cependant si certaines tâches de l'activité concernent tous les apprenants, ces dernières doivent être multi-instanciées en parallèle et attribuées à chaque apprenant comme pour le niveau 1.
APPRENANT	3 Scénario global initialisé par chaque apprenant. De ce fait, pas de multi-instanciation nécessaire. Cependant les tâches de gestion globale effectuée par l'enseignant (ex : Poster un document ressource) ne peuvent être incluse ce qui peut poser problème pour certains scénarios.	4 Une activité exécutée par un apprenant lorsque cette dernière est disponible pour lui. Le workflow peut être lancé plusieurs fois par l'apprenant tant qu'il y a accès. C'est le cas d'implémentation le plus simple car il correspond le plus à la vision d'entreprise de BPMN. Cependant cette vision centrée sur un apprenant rend délicat les modélisations de scénarios collaboratifs.

Tableau 4 : Perspective et niveau de granularité lors de la modélisation de workflow BPMN

2.3 Une plateforme pour l'exécution de scénarios.

Afin d'opérationnaliser notre démarche de conception avec BPMN, nous proposons d'expérimenter l'utilisation d'une infrastructure basée sur un moteur de processus BPMN pour l'exécution de workflow pédagogique. Un environnement d'apprentissage virtuel que nous pourrions qualifier de « E-learning Workflow-based Platform ». L'idée est d'expérimenter la faisabilité/utilité d'une telle technologie en éducation. Il n'existe évidemment pas à l'heure actuelle d'outil spécialisé en éducation pour sa mise en œuvre. Nous utiliserons donc les outils issus de l'industrie IT pour le développement de processus métier : les BPMS (chapitre 4).

Les BPMS sont des infrastructures de développement sophistiqués destinés à être utilisés par des analystes métiers passant ensuite le relais à des développeurs logiciels. Ils sont conçus pour mettre en œuvre des démarches d'ingénieries complexes pouvant assurer la sécurité de données sensibles (Ex : milieu bancaire) ou soutenir des charges intensives d'opérations. Nous ne recommandons pas de tels outils pour un usage en éducation (sauf dans le cas de grands projets de formation impliquant des équipes de développement). Toutefois, nous soutenons que le développement d'outils de même nature mais aux possibilités plus restreintes, adaptés aux besoins de l'éducation et plus simple dans leur utilisation serait possible et intéressant. À défaut, nous utiliserons dans le cadre de ce mémoire le BPMS « BONITA BPM 6 » proposant une solution gratuite, fonctionnelle, open-source et relativement simple à utiliser, en comparaison à la concurrence (Bonita reste toutefois un outil professionnel nécessitant des connaissances en programmation et ingénierie des systèmes d'information).

Nous avons pu déployer un portail web installé sur un serveur d'application Java en local. Ce portail est conçu à partir du BPM Portal de Bonita et intègre un moteur de gestion de processus. Dans notre optique, cette plateforme devra nous permettre d'exécuter les processus, gérer les utilisateurs et leurs droits ainsi que de suivre le déroulement des processus via des indicateurs statistiques. Nous avons intitulé cette plateforme « E-Flow ».

Pour les besoins de nos tests nous avons créé une organisation fictive se composant de groupes et sous-groupes d'utilisateurs fictif. Les comptes ont ensuite été créés manuellement depuis un compte administrateur et assignés à un groupe. Il est à noter qu'un utilisateur externe ne peut pas créer de compte sur la plateforme. Son compte doit être créé soit manuellement soit automatiquement via une synchronisation LDAP (Lightweight Directory Access Protocol) permettant de récupérer l'annuaire d'utilisateurs déjà existant au sein de l'organisation.

Un workflow lancé sur E-flow doit pouvoir être « mappé » avec les utilisateurs existants sur la plateforme. Si un diagramme BPMN contient trois couloirs, chacun de ces couloirs doit pouvoir correspondre à un utilisateur ou groupe d'utilisateur définis. Les groupes sont hiérarchiques. Ainsi une tâche attribuée à un groupe peut être effectuée par tous les acteurs de chaque sous-groupe. Ci-contre l'organisation hiérarchique des groupes utilisés sur la plateforme pour tester la distribution des tâches au sein d'un workflow pédagogique.

Organisation
Équipe Pédagogique
<ul style="list-style-type: none"> - Enseignants - Assistants
Étudiants
<ul style="list-style-type: none"> - Promotion 1 - Promotion 2 - Promotion 3
Service IT (Help Desk)
<ul style="list-style-type: none"> - Employé - Support général - Support avancé

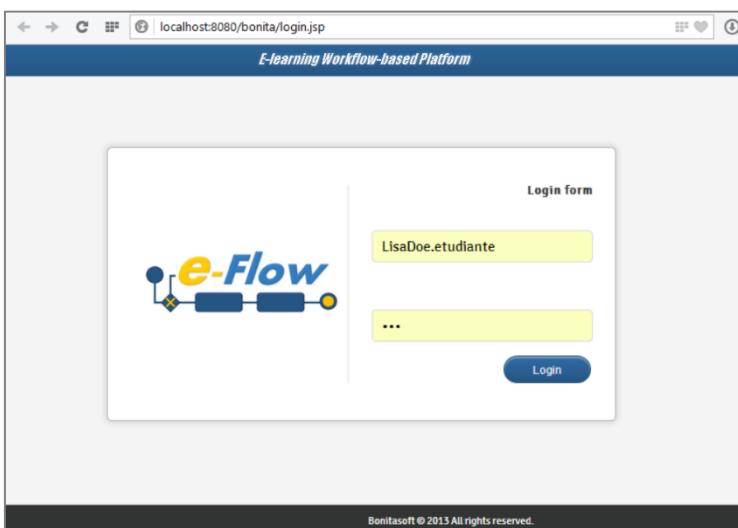


Figure 37: Écran de connexion à la plateforme E-flow

moment via un bouton dédié. En mode utilisateur (par défaut lorsqu'il se connecte) il effectue les tâches qui lui sont attribuées comme n'importe quel autre utilisateur du dispositif. En mode administrateur (figure 37), il peut accéder à l'interface d'administration lui permettant :

- D'installer des workflows créés avec le Bonita Studio et de relier les groupes d'utilisateurs au workflow (onglet gestion Apps).
- De créer et gérer les utilisateurs et leurs groupes (onglet Organisation).
- De consulter ou générer des rapports sur les données de formation (onglet métrique).
- De gérer les droits des utilisateurs (onglet configuration)

Figure 38 : Portail E-flow en mode administrateur- installation d'un scénario

Au point suivant nous testerons la modélisation de scénarios pédagogiques avec BPMN et leur possible exécution sur E-Flow

Le design général du portail web a été créé par modification des feuilles de style CSS ou LESS²² de Bonita. Les différents acteurs peuvent se connecter via un écran classique de connexion. Ils sont alors identifiés comme un utilisateur défini de l'organisation assigné à un groupe. Le moteur de workflow peut ainsi leur affecter les tâches telles que définies dans les scénarios BPMN.

Pour un administrateur l'interface du portail se compose de deux modes différents (mode utilisateur et mode administrateur).

Il peut basculer de l'un à l'autre à tous

²² Langage dynamique de génération de feuilles de style. Permet l'ajout à CCS des mécanismes de variables, fonctions, opérateurs.

3. ÉTUDE DE CAS

Afin d'aborder les potentialités et difficultés d'une approche de conception avec BPMN 2.0. Nous avons effectué différents tests plus ou moins formels de modélisation et d'implémentation. Cette exploration visait différents objectifs :

- Mettre en pratique la modélisation avec les éléments BPMN et le cadre de conception proposé au point 2.1.
- Tester les facilités et/ou difficultés des différentes tâches d'implémentation pour la mise en œuvre de ces scénarios via un moteur d'exécution de workflow.
- Comprendre ce qu'une approche de conception avec BPMN permet ou non de faire.

Les observations issues de ces tests seront en grande partie reprises en conclusion. Nous présentons simplement ici deux des tests afin d'illustrer la démarche et son potentiel. Le premier illustre l'implémentation complète d'une petite activité de correction très simple. Le second développe un scénario complet d'écriture de récit collaboratif en situation hybride. Il se compose de différents workflows de niveau 2 orchestrés par un workflow de niveau 1. Il représente un scénario complexe orchestrant différentes activités sur différentes itérations dans une perspective socioconstructiviste.

3.1 Activité simple de correction.

Afin d'illustrer le principe de conception nous présentons ici un petit workflow simple. Nous nous positionnons dans une perspective d'implémentation de niveau 2 avec initialisation par un étudiant. L'objectif est d'implémenter une petite application autonome pour la discussion et l'amélioration de travaux étudiant. Cette petite application peut être démarrée manuellement par un acteur du dispositif ou peut être intégrée à un scénario plus complexe tel que présenté au point suivant. Nous avons modélisé un petit workflow permettant la circulation automatisé d'un document entre différents acteurs jusqu'à être jugé valide pour la publication finale. De manière simple l'idée première aurait pu se représenter de la façon suivante :

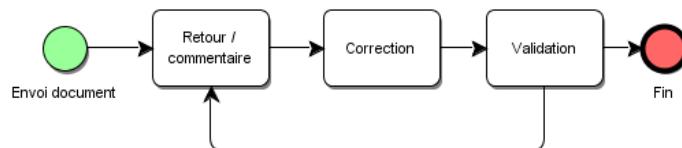


Figure 39: Modèle simple processus de correction

Ce processus a ensuite été enrichie et précisé afin d'en proposer une version exécutable par le système. Nous avons premièrement précisé la situation d'application, les acteurs impliqués et leur rôle respectif. Nous avons choisi une situation fictive de type académique comprenant enseignants, assistants et apprenants. L'idée est de fournir un système non linéaire permettant un bouclage infini entre ces trois acteurs tant que le document ne convient pas aux exigences. Dans notre scénario :

- **L'apprenant** : Produit le document, transmet le document à l'assistant, effectue les corrections nécessaires.
- **Les assistants** : Sont chargés des premières lectures et des prises de décision. Ils peuvent notamment demander des corrections et revoir le document, demander des corrections et transmettre à l'enseignant pour validation, décider de publier le document tel quel, etc.
- **L'enseignant** : Décide si le document peut être publié ou publié après quelques corrections mineures ou enfin nécessite des corrections importantes impliquant un retour point de départ.

Après différents tests notre modélisation s'est arrêtée sur le diagramme BPMN de la figure 40.

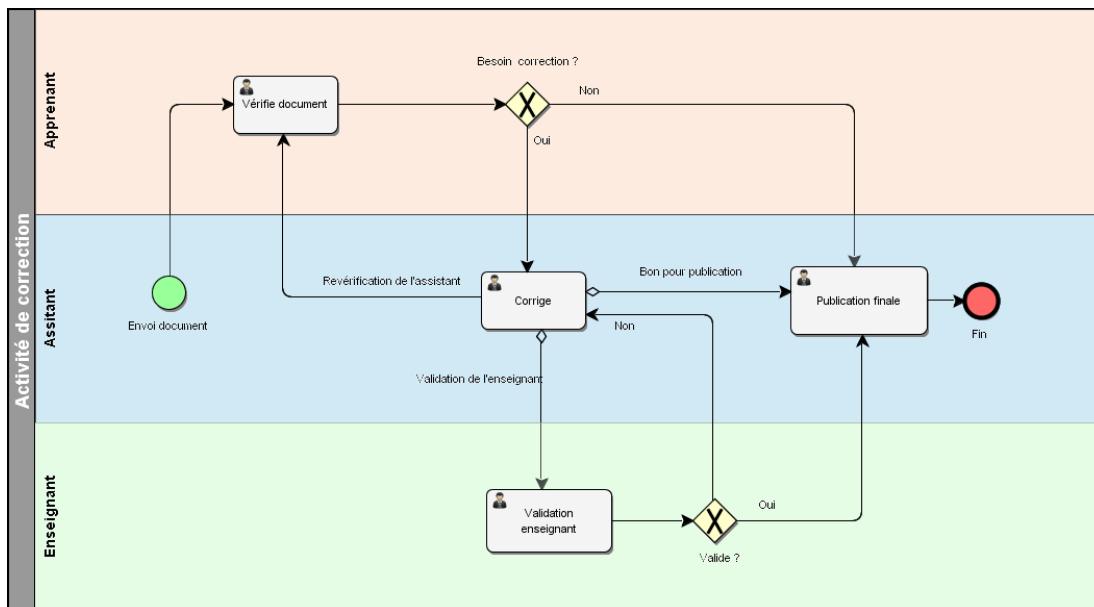


Figure 40: Diagramme BPMN final activité de correction

L'implémentation a suivi la procédure présentée en annexe 10. Des variables booléennes conditionnent le parcours en prenant les valeurs « True » ou « False » en fonction des choix des participants dans les formulaires. L'application une fois exportée en fichier .bar peut être installée sur E-flow et initialisée par n'importe quel étudiant voulant démarrer un cycle de correction de son travail avec l'aide de l'équipe pédagogique. Cette activité peut donc être lancée manuellement par chaque apprenant mais peut aussi être utilisée dans le cadre d'un scénario de niveau 1. Auquel cas l'activité de correction sera initialisée par le système à un moment donné et selon les conditions définis dans le scénario BPMN niveau 1 orchestrant les activités.



Figure 41: Activité de correction disponible sur E-Flow pouvant être démarré par un étudiant.



Figure 42: Nouvelle tâche disponible pouvant être réalisé par un assistant

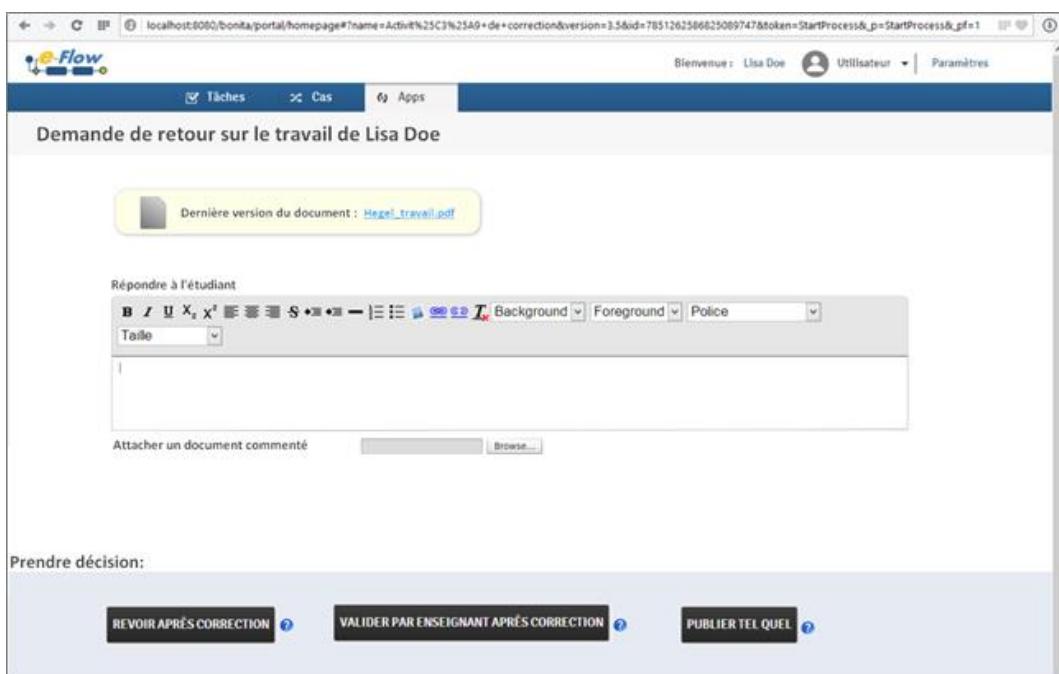


Figure 43: Réalisation de la tâche par un assistant

3.2 Écriture de récits collaboratifs en contexte hybride.

Intentions pédagogiques

Notre scénario pédagogique invoque la construction d'un récit en communauté. Les étudiants construisent (seul ou à deux) à distance des suites à un début d'histoire fourni par l'enseignant et votent chaque semaine pour la meilleure. Le scénario se focalise sur la pratique réflexive de l'apprenant et la confrontation de connaissances aux points de vue d'autrui. À chaque itération le dispositif enjoint l'apprenant (par des questions) à réfléchir sur la manière dont il a abordé l'écriture de l'itération précédente et à faire un petit commentaire sur l'écriture en cours.

Notre scénario s'ancre dans une perspective socioconstructiviste de l'apprentissage que nous pourrions énoncer selon les principes de Doolittle (1999) de la manière suivante :

1. *Présenter une activité complexe et authentique.* La production d'un récit commun représente le résultat tangible issu d'un travail collaboratif global sur plusieurs semaines.
2. *Procurer des interactions sociales.* À chaque itération l'apprenant doit argumenter, lire l'argumentation de ses paires et la discuter dans la perspective concrète d'un vote « éclairé ». L'interaction utilise les technologies de communication.
3. *Fournir un contenu et un savoir-faire signifiants et proche des acquis de l'élève.* Le scénario s'ancre dans le programme d'un cours de français de niveau lycée. Il représente une application originale et assez ludique d'aborder l'expression écrite et ses principes (architecture et construction de récit, style littéraire, figures etc.)
4. *Bénéficier d'une évaluation formative continue.* Chaque itération fournit un feedback issu de la communauté. L'enseignant peut aussi effectuer un retour en classe ou en ligne sur les réalisations de la semaine.
5. *Être responsable de son apprentissage.* L'apprentissage de l'apprenant dépend de lui-même et non de son enseignant. Cet apprentissage nécessite un engagement profond de la part de l'apprenant dans la participation au travail de la communauté.
6. *Les enseignants sont d'abord des guides et des agents facilitants de l'apprentissage.* L'enseignant n'est pas un instructeur. Il propose plutôt à l'élève des expériences aptes à susciter des acquisitions significatives.

7. *Revoir des contenus et les présenter selon diverses perspectives.* L'étudiant à travers la confrontation des travaux à la vision de ses paires apprend à percevoir la réalité selon divers points de vue afin de construire sa propre vision de cette réalité perçue (formalisé par l'acte final du vote).

Aborder un tel scénario en contexte hybride nécessite une technologie de support permettant sa mise en œuvre organisationnelle mais aussi pédagogique. La réussite (création d'une réelle compétence d'écriture chez l'apprenant) ne peut être atteinte qu'au prix d'une scénarisation précise des activités. Le socioconstructivisme postule avant tout que l'apprentissage survient lorsque des individus tentent de coordonner des centractions cognitives divergentes dans la perspective d'une action commune concrète. Les conditions nécessaires à ce processus ne vont pas de soi et résultent d'une scénarisation visant à provoquer des apprentissages.

De manière globale notre modélisation et son implémentation doivent faire démarrer le travail, rendre l'enseignant disponible au moment opportun, gérer le temps, faire participer les apprenants, susciter le questionnement, confronter l'apprenant au travail des autres, gérer le flux documentaire etc.

Modélisation

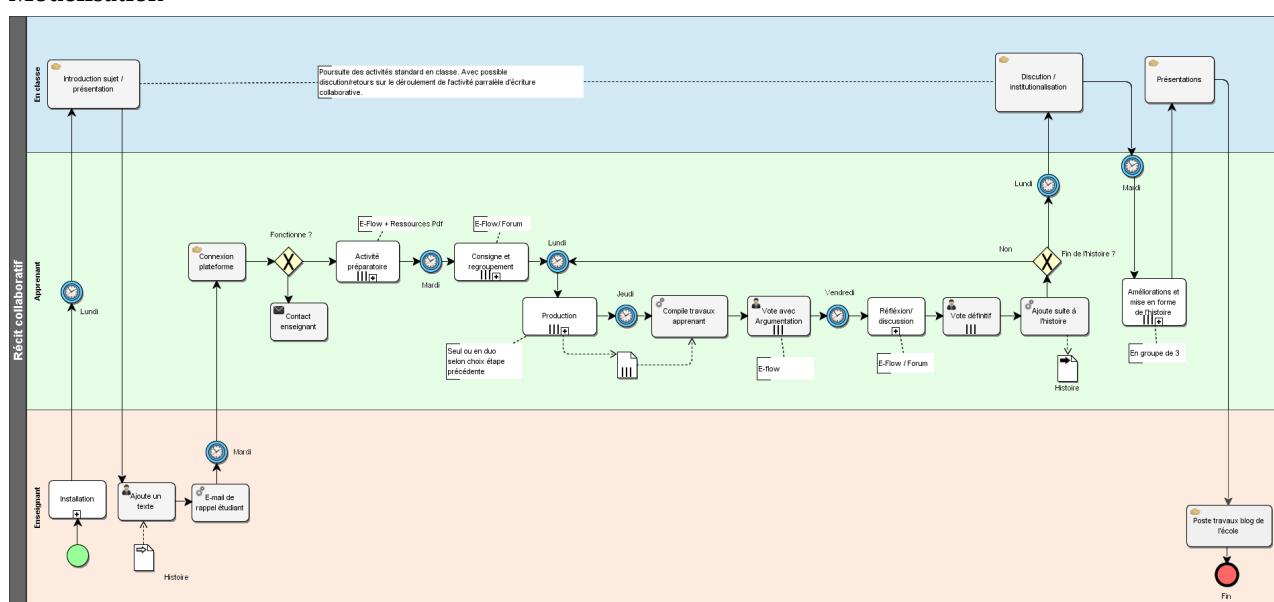


Figure 44: Scénario récit collaboratif (niveau 1) - Grand format voir annexe 9

Les différentes phases du scénario consistent à (1) présenter l'activité aux étudiants, introduire le sujet par une activité préparatoire, (2) introduire les consignes, s'assurer que tous les apprenants ont bien compris les directives, former des groupes ou non, (3) proposer une suite, (4) voter pour une suite et argumenter son choix, (5) lire les argumentations, les discuter, (6) voter définitivement, (7) réitérer les phases précédentes jusqu'à fin de l'histoire en ajoutant des étapes permettant d'interroger l'étudiant sur sa pratique à chaque itération, (8) institutionnaliser, (9) concrétiser le travail par la production d'un produit finalisé et esthétique (motivation des élèves aimant le graphisme), (10) présenter aux autres, (11) publier sur le blog de l'école (motivation).

La modélisation se situe à un niveau 1 et décrit l'enchaînement des différentes étapes selon les acteurs, la temporalité, les ressources et outils utilisés. Chaque appel d'activité (en blanc) est multi-instancié en parallèle et attribué à chaque apprenant. Ces activités font alors références à un workflow de niveau 2 (voir ci-après) décrivant le parcours de l'apprenant.

Le scénario donne une vision globale du déroulement des tâches des acteurs mais aussi du système (ex : compilation des travaux étudiants, attribution d'un étudiant à un groupe). Les tâches et activités n'impliquant pas d'interaction avec le système E-flow sont tous simplement ignorées et sautées par le système durant l'exécution. Des évènements de temporalité permettent alors de gérer et coordonner le flux. Les attributs

définissant ces temporalités ne sont pas fixes mais récupèrent la valeur d'un paramètre devant être complété par l'enseignant durant l'installation. Durant les itérations les temps sont définis comme des cycles (ex : tous les lundis à 8h) permettant d'appliquer le même schéma chaque semaine.

La modélisation représente un workflow global lancé une seule fois par l'enseignant pour tout le dispositif. Chaque appel d'activité est ensuite décrit par un diagramme au niveau 2.

À titre d'exemple la figure 45 ci-après est un diagramme niveau 2 concernant l'activité « production ». Il décrit le parcours instancié pour chaque apprenant à cette étape du dispositif. Ce parcours se présente différemment si (1) l'apprenant a choisi d'effectuer le travail seul ou en groupe durant l'acte précédent (figure 46) et (2) si l'apprenant se trouve à la première itération du processus ou les suivantes. En cas de retard important à une date fixée un signal se déclenche et envoie un avertissement à l'enseignant. Ce dernier l'informe de la situation de l'étudiant et lui offre différents choix de régulation (figure 44). Chaque activité appelante est définie à un niveau 2.

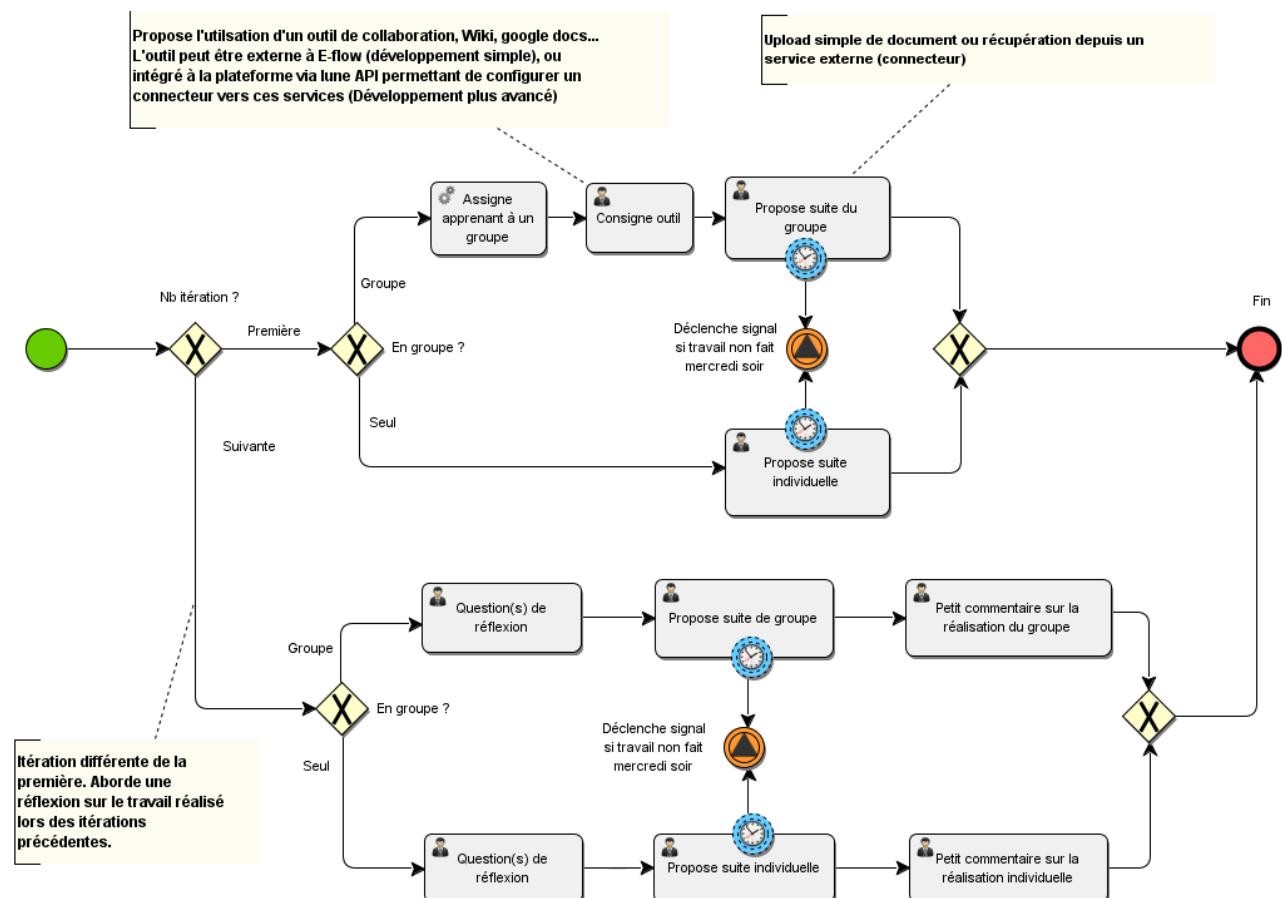


Figure 45: Description activité « production » (niveau 2)

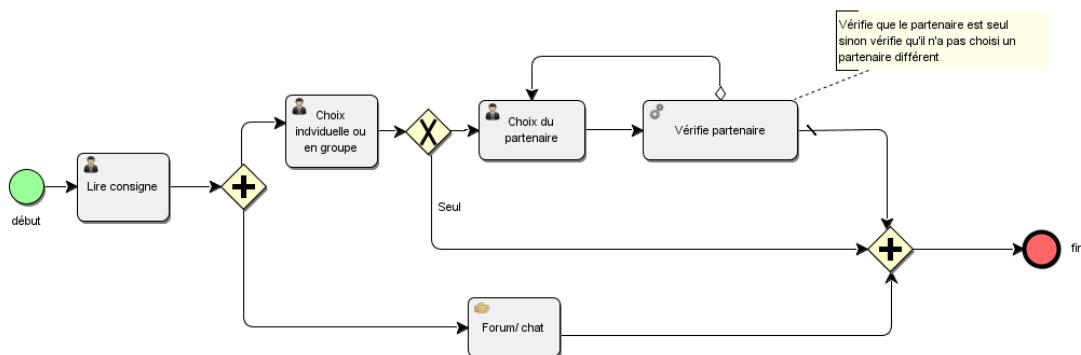


Figure 46: Description activité « consigne et groupement » (niveau 2)

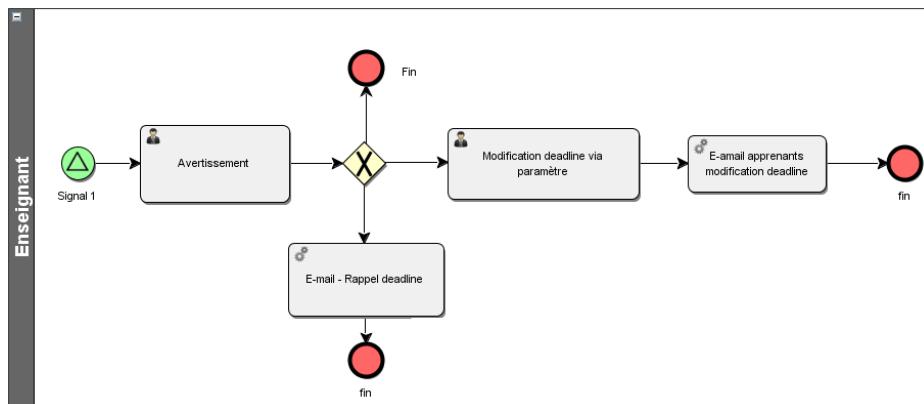


Figure 47: Processus de régulation enseignant déclenchée via signal.

Implémentation

L'ensemble du scénario n'est pas fonctionnel mais plusieurs sections ont pu être implémentées avec succès (Formulaires, connecteur e-mail, intégration Google docs « simple » via Iframe, gestion de document, étape de vote, déclenchement sur signal...).

- Concernant les diagrammes de niveau 2, ces derniers ont été implémentés de la même manière que l'activité de correction présentée au point précédent. Les étapes non implémentées sont celles nécessitant l'écriture de code Java pour la définition d'opérations spécifiques non prises en charge (ex : compilation de travaux par le système).
- Concernant le diagramme de niveau 1, la tâche s'avère plus complexe. La difficulté majeure concerne la multi-instanciation des activités. En éducation chacune des tâches d'apprentissages doit être réalisée par chacun des étudiants ce qui se révèle très différent des approches classiques de workflow en entreprise. En entreprise lorsque une tâche nécessite d'être réalisée elle est envoyée à l'ensemble des acteurs concernés. Lorsqu'un acteur effectue la tâche, celle-ci n'est plus à faire et disparait donc du flux d'activité. En éducation la démarche est différente. Une tâche doit être instanciée autant de fois qu'il y a d'étudiants ou groupes d'étudiants. Chacune de ces instances doit donc être attribuées dynamiquement à chacun des acteurs. Le BPMS Bonita n'offre pas de moyen simple de réaliser une telle opération, cette dernière doit donc être réalisée par programmation (script Groovy ou Java).

Concernant la gestion des données de formation nous n'avons pu tester la définition de KPI personnalisé non disponible avec notre licence. Cependant nous disposons de modèles de rapports inclus par défaut permettant de générer des rapports sur le statut des tâches en cours, terminées, échouées, le temps impliqué pour la réalisation de chaque tâche etc.

CHAPITRE V – MODÉLISATION DE SCÉNARIO PÉDAGOGIQUE AVEC BPMN

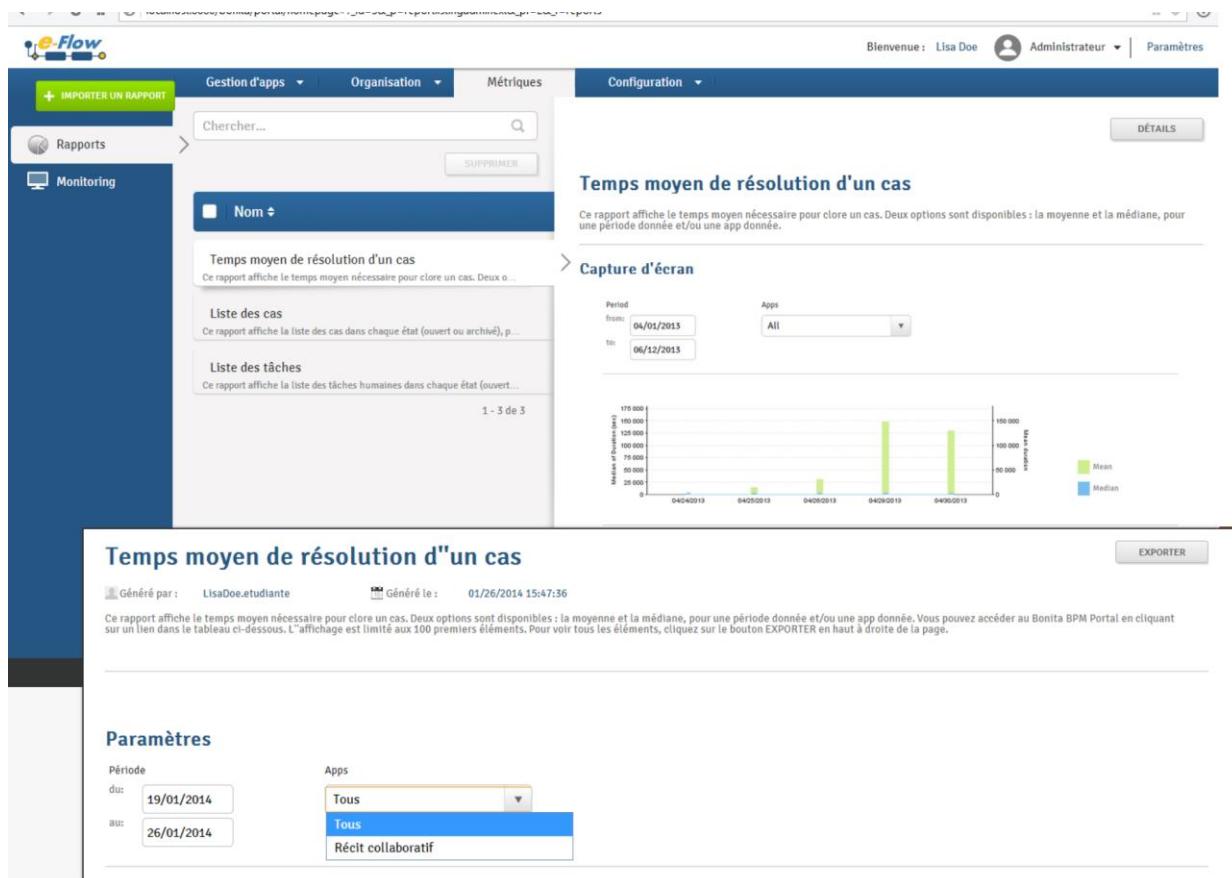


Figure 48: Aperçus de l'interface d'E-flow pour la gestion de données de formation

CONCLUSION ET DISCUSSION

Durant ce chapitre nous avons pu nous confronter à la conception de dispositifs basés sur (1) des scénarios pédagogiques formalisés en BPMN, (2) une plateforme d'exécution de workflow et (3) une architecture orientée services. Nous avons défini une palette d'éléments graphiques apte selon nous à représenter des « workflows pédagogiques » en accord avec le cadre théorique développé au long de ce mémoire. Cette dernière nous a permis de définir un flux de travail non linéaire représentant l'ordre conditionnel et/ou temporel d'exécution des tâches par les différents acteurs impliqués dans l'activité ainsi que les ressources utilisées, transformées ou produites lors du processus. Notre modélisation s'est appuyé sur :

- Des diagrammes à visée communicationnelle progressivement affinés vers une perspective d'implémentation ;
- Une décomposition hiérarchique sur deux ou trois niveaux de granularité ;
- Une définition préalable des conditions d'initiation du processus et d'instanciation des tâches.

L'implémentation a été réalisée avec un BPMS permettant de lier les éléments graphiques à des variables et script ainsi que des connecteurs simplifiant le protocole de communication à des services externes (via API). Une plateforme intégrant un moteur de gestion de workflow nous a permis d'exécuter les scénarios selon les règles conditionnelles et temporelles préalablement établis. À l'issu de cette exploration nous percevons le potentiel que revêt BPMN 2.0 pour l'éducation mais aussi les difficultés et limites.

Au niveau de la modélisation nous constatons que BPMN est particulièrement efficace pour représenter des scénarios :

- nécessitant qu'un apprenant emprunte des chemins différents selon certaines conditions ;
- devant présenter des retours ou itérations ;
- nécessitant le déclenchement d'activités exceptionnelles en fonction d'évènements ;
- présentant un fort aspect coopératif (partage et distribution de tâches entre acteurs) ;
- reposant sur une pédagogie constructiviste ;

À contrario le langage nous a posé des difficultés pour représenter des scénarios :

- contenant des phases de regroupement d'apprenants ;
- impliquant la distribution d'une tâche commune à des acteurs ou des rôles différents ;

Si nous avons tout de même toujours trouvé un moyen de représenter un scénario avec BPMN nous pouvons constater qu'il n'est pas toujours clair de savoir quelle est la meilleure manière de représenter un scénario présentant un fort aspect collaboratif, surtout lorsque l'on se pose la question au niveau de la sémantique d'exécution en jeu derrière le diagramme. Ce problème est selon nous intrinsèquement lié au concept même de workflow et sa dynamique axée sur « un partage distributif des tâches » et non sur « une réalisation commune de celles-ci».

Au niveau de l'implémentation nous avons pu mettre en exergue la faisabilité d'une conception d'EIAH simple selon une approche IDM avec BPMN. Cette dernière s'est révélée apte à soutenir nos intentions pédagogiques et nous a permis d'éprouver un cycle constructif (« modélisation – test d'exécution ») nous enjoignant continuellement à améliorer nos modèles et préciser nos intentions. Les perspectives de connexion à des services externes et d'intégration aux infrastructures déjà existantes en organisation semblent nombreuses et mériteraient d'être investiguées de manière plus approfondie. En outre, la possibilité d'associer modélisation et analyse de données de formation par le biais de KPI attachés aux diagrammes BPMN offre un potentiel intéressant en Learning Analytics. Toutefois l'exploration des perspectives d'implémentation de scénarios via les BPMS nous permet d'émettre les difficultés techniques suivantes (logiquement liées aux problèmes collaboratifs identifiés ci-dessus pour la modélisation).

1. *Attribution collective.* La logique de workflow derrière BPMN conçoit un flux de tâches d'individu à individu

l'implémentation de scénarios à un niveau macro (type niveau 1) gérant une cohorte d'étudiant devant tous réaliser une même tâche, est relativement complexe et passe par la réalisation manuelle d'un script (filtre d'acteur) gérant dynamiquement la multi instantiation des tâches. De manière simple ce script traduit la fonction suivante « instancier autant de fois que le nombre d'étudiants identifiés et attribuer chaque instance à chaque identifiant ».

2. *Attribution à un groupe* : De la même manière l'attribution automatique d'un étudiant dans un groupe n'est pas une fonctionnalité par défaut et passe par un script; Toutefois lorsque le nombre d'étudiant reste « raisonnable » il est possible de le faire manuellement par un administrateur sur la plateforme.
3. *Multi-instantiation de groupe* : similaire au point 1 mais avec les groupes. Instancier autant de fois que de sous-groupes dans un groupe identifié et attribuer à chaque sous-groupe.

Enfin nous soulevons des interrogations d'ordre global quant aux problématiques suivantes :

1. *Vision globale*. Une fois dans le dispositif un acteur n'a pas de vision globale du workflow. L'apprenant réalise les tâches mais n'a pas de réelle vision de la poursuite de son parcours. Un workflow complexe d'implémentation ne peut être fourni à un acteur du dispositif toutefois une vision simple des grandes étapes du parcours pourrait lui être utile. Lors d'une conférence BonitaSoft (Genève, septembre, 2013), cette problématique apparaissait dans plusieurs questions. Les conférenciers assurèrent alors qu'une solution était actuellement à l'étude et semblait en passe de trouver un « compromis » permettant l'affichage d'une vision très simplifiée du diagramme à destination de l'utilisateur final.
2. *Adaptation à la volée* : Cette problématique est centrale et commune à tout langage de modélisation planifiant l'activité humaine (source de nombreux imprévus). Nous apporterons à ce titre deux pistes de solutions. La première étant la possibilité d'anticiper les problèmes durant la modélisation. De ce point de vue BPMN est un langage relativement efficace (comparativement à d'autre) pour déclencher des chemins exceptionnels en fonction d'évènements identifiés. En outre, les modèles peuvent être améliorés à chaque cycle de réingénierie pour parer aux difficultés du cycle précédent. La seconde consiste à utiliser des variables de processus récupérant la valeur d'un paramètre (depuis 2013), la valeur du paramètre étant entré par un utilisateur et modifiable à tout moment par ce dernier depuis un panneau d'administration. Enfin, cette problématique étant absolument centrale pour BPMN 2.0, des solutions techniques sont en passe de réellement voir le jour pour modifier « à chaud les workflows ». Ainsi Bonita BPM 6 annonce la possibilité dans ses versions sous licences de pouvoir déclencher des tâches non prévues par le workflow et de les assigner à des utilisateurs, à terme cette solution devrait permettre le déclenchement de sous-processus complets.

CONCLUSION ET PERSPECTIVES

Dans le cadre de ce mémoire nous avons exploré la possibilité d'utiliser BPMN 2.0 en tant que langage graphique de modélisation pédagogique. En étudiant une technologie issue de l'ingénierie des processus métier, notre démarche contribue à ouvrir l'horizon des technologies éducatives et vise plus globalement à soutenir une réflexion sur l'avenir des environnements numériques d'apprentissage et la place des intentions pédagogiques dans les processus de conception.

Ce travail nous permet de présenter le potentiel futur et les perspectives présentes que représente BPMN pour l'ingénierie de formation. Il nous permet aussi d'entrevoir les enjeux à relever par le langage et les limites d'une approche par workflow en pédagogie.

RÉSUMÉ ET BILAN

Au cours de ce mémoire nous avons pu constater un intérêt croissant en ingénierie pour des techniques et méthodes permettant de formaliser, communiquer et partager des scénarios d'apprentissage. Notre exploration des technologies soutenant les pratiques d'ingénierie (standards e-learning, script CSCL, différents LMP visuels) a mis en exergue le manque de solutions réellement opérationnelles pour décrire des scénarios pédagogiques riches²³ de manière humainement compréhensible, formelle, standard et exécutable (au moins partiellement) dans des environnements de formation. Outre les caractéristiques pédagogiques intrinsèques à un LMP, nous considérons que pour avoir une chance de se diffuser un langage devrait [1] être compréhensible par des acteurs d'horizons professionnels différents, [2] être opérationnel (outils), [3] faciliter l'implémentation (rationalisation de l'effort de modélisation) et [4] être normalisé et lié à un standard (échange entre concepteur, durée de vie etc.).

Notre exploration nous a également permis d'aborder la question de la conception des EIAH. La complexité de ces derniers implique souvent leur développement par des personnes différentes de celles qui les utiliseront (les formateurs, enseignants, ingénieurs pédagogiques). BPMN étant un langage de développement au cœur de l'ingénierie des processus métier, nous nous sommes donc interrogés sur la faisabilité d'une approche dirigée par les modèles pour la conception d'EIAH afin d'introduire plus fortement le pédagogue dans les processus de conception. Les principes de l'approche IDM et de l'architecture SOA sont alors apparus comme un sujet d'intérêt pour concevoir des EIAH modulaires et interconnectables, générés à partir de modèles capturant les intentions pédagogiques. L'idée étant de fournir des dispositifs techniques s'adaptant au design pédagogique et non le design pédagogique s'adaptant aux possibilités du dispositif technique.

Afin de tester les possibilités de mise en œuvre de notre approche nous sommes intéressés aux caractéristiques du langage BPMN et à ses outils de développement. Concernant BPMN, le langage est graphique et abstrait (neutralité), standardisé et apte à représenter des activités humaines complexes. Le langage définit une sémantique d'exécution et il est largement diffusé dans les formations académiques et professionnelles du personnel se destinant à des fonctions managériales ou de services IT. Notre exploration des outils nous a permis d'aborder des solutions logicielles nombreuses et diverses, allant d'outils simples à des suites professionnelles de développement (BPMS). Nous avons utilisé des BPMS et avons pu nous rendre compte d'outils puissants, extensibles et innovants permettant d'assurer toutes les phases de développement d'un cycle d'ingénierie

²³ Nous entendons par là des scénarios invoquant une pédagogie active d'inspiration plutôt (socio)constructiviste. De tels scénarios se traduisent au niveau technique par des parcours aux multiples embranchements et non linéaire. La technologie doit pouvoir soutenir et accompagner divers processus de découverte, de construction, de partage, de discussion, de confrontation, d'intégration, de renforcement etc.

CONCLUSION ET PERSPECTIVES

(modélisation, implémentation, test, déploiement en ligne, gestion du déroulement, analyse et évaluation.). De surcroit ces outils tentent de se démarquer sur un marché très concurrentiel par une politique visant à rendre les outils plus simples, intuitifs, et limitant au maximum les phases de développement de code. En pratique ces outils se révèlent toutefois encore très techniques et leur orientation « entreprise » entraîne un manque de fonctionnalités « pré-intégrées » utiles pour l'éducation. De plus nous avons pu constater que le standard BPMN 2.0 est implanté de manière inégale, entraînant par conséquent une interopérabilité variable entre systèmes. La spécification est cependant très récente (2011-2012) et des améliorations se profilent.

Enfin nous nous sommes confrontés au cours du dernier chapitre de ce mémoire à la modélisation et l'implémentation effectives de scénarios pédagogiques riches modélisés avec BPMN. Nous avons utilisé un BPMS open source avec une licence gratuite (assez complète pour nos tests) et les principes de l'approche BPM, de la démarche IDM et de l'architecture SOA. Nous avons ainsi pu évaluer notre approche basée sur (1) des scénarios pédagogiques formalisés en BPMN, (2) une plateforme d'exécution de workflows et (3) une interface de gestion et d'analyse des données de formation. Nous avons proposé une palette d'éléments graphiques apte selon nous à représenter des « workflows pédagogiques » en accord avec le cadre théorique développé au long de ce mémoire (théorie de l'activité, richesse des scénarios). Cette palette peut toutefois être enrichie avec n'importe quel autre élément défini dans la spécification BPMN 2.0. Ces éléments nous ont permis de décrire un flux de travail non linéaire représentant l'ordre conditionnel et/ou temporel d'exécution des tâches par les différents acteurs impliqués dans l'activité ainsi que les ressources utilisées, transformées ou produites lors du processus. Notre modélisation s'est appuyée sur :

- Des diagrammes à visée communicationnelle progressivement affinés vers une perspective d'implémentation ;
- Une décomposition hiérarchique sur deux ou trois niveaux de granularité ;
- Une définition préalable des conditions d'initiation du processus et d'instanciation des tâches.

L'implémentation a été réalisée avec un BPMS permettant de lier les éléments graphiques à des variables et scripts ainsi que des connecteurs simplifiant le protocole de communication à des services externes (via API). Une plateforme intégrant un moteur de gestion de workflow nous a permis d'exécuter les scénarios selon les règles conditionnelles et temporelles préalablement établies.

DÉFIS À RELEVER POUR LA MISE EN ŒUVRE DE BPMN EN ÉDUCATION

Le premier défi que nous identifions est trivial mais nécessaire et concerne la diffusion du langage en éducation. BPMN à la réputation d'un langage « administratif » et « technique », destiné au monde de l'entreprise et à ses équipes IT. Son usage dans une perspective éducative est en soi une catachrèse et doit s'accompagner de travaux, méthodes et exemples théorisant sa pratique et confirmant son utilité en technologies éducatives. En outre, la logique de « partage » en jeu derrière la modélisation de scénarios pédagogiques ne peut porter ses fruits qu'avec une large adoption des outils et du standard BPMN (le partage étant toujours assuré par une minorité d'utilisateur).

Le second défi découle du précédent et a trait aux principes de l'architecture SOA. La logique des « connecteurs » employés par de nombreux BPMS est très intéressante et pourrait avantageusement être mise à profit dans le domaine éducatif. Actuellement trop peu de connecteurs sont développés dans une perspective éducative afin de faciliter la connexion aux outils web 2.0, plateforme LMS et services divers utilisés en technologies éducatives. Des logiciels comme Bonita disposent d'une communauté de partage en ligne permettant de mettre à la disposition des autres utilisateurs des « connecteurs » pouvant simplifier les projets de développement. Nous soutenons qu'une logique similaire est envisageable dans le monde de l'éducation.

Le troisième défi concerne les outils en eux-mêmes. Les BPMS sont selon nous des infrastructures puissantes et déjà opérationnelles actuellement pour des projets de formation intégrant des équipes de développement.

CONCLUSION ET PERSPECTIVES

Cependant un BPMS spécialisé pour l'éducation (un concept de Educational Process Management System - « EPMS ») nous semble nécessaire pour un usage efficace et dans des projets plus modestes. Dans cette perspective fictive, le travail exploratoire réalisé au cours de ce mémoire nous permet d'esquisser les recommandations suivantes :

- L'EPMS peut être plus simple et moins puissant. Un BPMS doit en effet, faire face à des contraintes opérationnelles non présentes en contexte éducatif (sécurité de données bancaires, charge de traitement, gestion d'activité pouvant relever d'un danger pour le personnel etc.)
- L'EPMS doit garder la logique des « connecteurs» mais intégrer des outils courants aux portails d'exécution. Les connecteurs sont garants de la flexibilité du système et soutiennent la créativité des concepteurs pédagogiques pouvant être libres d'utiliser les services correspondant à leurs besoins. Toutefois l'intégration de quelques outils simples tels qu'un chat, forum ou wiki permettrait d'exécuter des scénarios pédagogiques dans des projets de formation plus modeste sans devoir gérer de connexion à des services externes.
- L'EPMS doit avoir une interface adaptée à des ingénieurs pédagogiques. Des utilisateurs avertis et expérimentés mais non techniques. Cela consiste surtout à rendre des rapports d'erreurs plus simples et à fournir des aides contextuelles plus nombreuses pour aider au paramétrage.
- L'EPMS doit intégrer par défaut des filtres d'acteurs plus puissant afin de s'adapter à la gestion du collectif propre au domaine éducatif. La logique de workflow en éducation implique des phases de regroupement et de réalisation des tâches par tous les étudiants et non un seul. (Voir chapitre 5)
- L'EPMS doit proposer certains moyens d'interactions avec des standards du E-learning (par exemple, import d'archive Scorm).

ENJEUX ET LIMITES D'UNE APPROCHE PAR WORKFLOW

Au cours de ce travail nous avons abordé la scénarisation sous l'angle d'un « workflow ». Dans nos scénarios modélisés avec BPMN les acteurs sont amenés à réaliser des tâches en utilisant des outils (traitement de texte, simulation...) et des services (forum, messagerie, chat...) selon un flux de travail pouvant être multiple, non-linéaire et selon un ordre conditionnel et/ou temporel défini par le concepteur. Nous avons eu l'occasion de montrer les correspondances entre cette approche et notre vision de l'apprentissage instrumenté dans laquelle les TICE visent à *soutenir*²⁴ des processus²⁵ d'apprentissage. BPMN est une notation graphique efficace pour modéliser et implémenter des scénarios impliquant des processus interactifs et des cheminement complexes et différentiés selon l'apprenant. D'une manière générale notre approche s'est révélée intéressante pour la création de dispositifs instrumentés basés sur une pédagogie constructiviste et tout particulièrement des scénarios de type approche par investigation ou par projet.

Cependant nous percevons aussi à l'issu de ce mémoire des difficultés concernant l'utilisation de BPMN, et plus généralement la logique de « workflow », au regard de la collaboration. La logique de workflow n'est pas *collaborative* mais *coopérative*. Cette différence est fondamentale et riche d'implication au niveau de la modélisation comme de l'implémentation (voir conclusion chapitre 5). Au niveau de la modélisation nous constatons qu'il n'est pas toujours clair de savoir qu'elle est la meilleure manière de représenter avec BPMN des phases de groupements d'étudiants ou des activités communes à différents acteurs et rôles. Au niveau de l'implémentation les outils de développement offrent rarement de fonctions simples pour assigner des tâches

²⁴ « *Soutenir* » car les technologies, tout comme les formateurs, ne pourront jamais que créer les conditions propices à l'apparition d'un apprentissage.

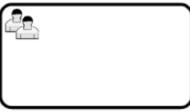
²⁵ « *Processus* » car « apprendre » est un processus cognitif complexe émergeant au cours d'activité de découverte, de construction, de partage, de discussion, de confrontation, d'intégration, de renforcement... La scénarisation a dès lors pour objectif structurer et organiser ces différentes activités en vue de provoquer des apprentissages identifiés.

CONCLUSION ET PERSPECTIVES

collectives ou assigner dynamiquement des utilisateurs à des groupes etc. BPMN étant une spécification récente, amenée à évoluer au cours des prochaines années, nous l'encoignions à une prise en compte de la dimension collaborative de l'activité humaine. Nous pourrions à titre d'exemple apporter les ajouts suivants à la spécification :

- Collaborative userTask

L'élément *collaborative userTask* définit une tâche effectuée en interaction avec un système de manière collaborative (ex : écriture collaborative). L'ajout d'un attribut *Actors associations* permet de spécifier le regroupement.

collaborative userTask	
Symbol	Attributes
	<i>id, name, renderings, implementation, resources, ioSpecification, dataInputAssociations, dataOutputAssociations, loopCharacteristics, boundaryEventRefs, ActorsAssociations.</i>

- Attribut *isCollective*

L'attribut *loopCharacteristics* fait appel à une classe support « *Sub-Class Supporting Classes* » définissant en BPMN les éléments *StandardLoopCharacteristics* dont le symbole est  et *MultiInstanceLoopCharacteristics* dont le symbole est  ou  (parallèle ou séquentielle). Dans une perspective éducative il pourrait être intéressant d'ajouter un attribut *isCollective* et un symbole dérivé  permettant de définir une tâche devant être réalisée par chacun des acteurs d'un couloir BPMN (multi-instanciation avec attribution automatique aux acteurs d'un couloir).

multiInstanceLoopCharacteristics		
Symbols	Attributes	Exemple
 /  /  / 	<i>id, isSequential, loopDataInput, inputDataItem, isCollective</i>	

- Intersection de couloir

Nous proposons enfin la possibilité d'ajouter une tâche, ou un appel d'activité à l'intersection de deux couloirs. Les modèles gagneraient en flexibilité, en permettant ainsi la représentation d'une action commune à différents rôles. Il est ainsi possible de représenter une activité réalisée en classe à l'intersection d'un couloir « enseignant » et « apprenants ». Bien que cela ne soit pas autorisé par le standard actuel, ce mécanisme est très attendu et se rencontre déjà sur certaines modélisations (voir exemple de Jiří Kolář et Tomáš Pitner, en Annexe 11).

- Emploi des couloirs dans les sous-processus

Nous avons pu constater que la norme BPMN n'est pas claire concernant la possibilité d'ajouter un bassin composé de différents couloirs dans un sous processus. En conséquence les outils de modélisation le permettent ou non. L'emploi de couloir dans les sous-processus est intéressant. À titre d'exemple, un sous-processus dans un couloir « étudiants » peut ainsi être scindé en couloirs « groupe1 » et « groupe2 ».

Enfin nous constatons, à titre d'ouverture au travail réalisé dans ce mémoire et au regard de notre expérience avec BPMN, des évolutions intéressantes concernant la problématique d'adaptation à la volée des workflows. C'est un enjeu de taille pour les processus d'entreprise et à fortiori pour la modélisation des dispositifs d'apprentissages. Le langage BPMN et les industrielles prennent à cœur cette problématique et proposent des mécanismes de plus en plus aboutis pour ajouter de la flexibilité aux workflows. Ainsi nous avons pu constater des fonctionnalités logicielles en phase de test permettant la modification en temps réel de paramètres impactant

CONCLUSION ET PERSPECTIVES

le cours du workflow ou la création de sous-tâches à la volée non prévus par le workflow initial. Ces évolutions laissent entrevoir de belles perspectives et pourraient bien aviver l'intérêt de l'ingénierie pédagogique pour les approches de conception par workflow en éducation.

Si « *le défi de concilier la scénarisation pédagogique normalisée et l'idéal des pratiques d'enseignement demeure quand même de taille parce qu'il s'adresse à des systèmes de valeurs encore assez éloignés* » (Henri, Compte et Charles, 2007). Les pratiques d'ingénierie sont et seront incontournables pour la formation de demain et doivent être acceptées et accompagnées par le pédagogue.

« *De même que cela n'a pas éliminé le cinéma d'auteur, cela n'empêchera pas l'engagement continual et original de l'enseignant dans les processus de production afin qu'il s'assure de l'adéquation entre ce qu'il propose et les besoins des apprenants.* » (*ibid*)

BIBLIOGRAPHIE

- Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I., Angel S. (1977). A Pattern Language, New York: Oxford University Press.
- Ausubel, D.P. (1960), The use of advance organizers in the learning and retention of meaningful verbal material. In *Journal of Educational Psychology*, 51, 267-272.
- Ausubel, D.P. (1968). *Educational Psychology: A Cognitive View*. New York: Holt, Rinehart & Winston.
- 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.
- Baron G.-L., « Learning design », *Recherche & formation*, n° 68, 2011, p. 109-120.
- Bertand, Y. (1979). *Les modèles éducationnels*. Montréal. Service pédagogique de l'Université de Montréal.
- Bertand, Y. & Valois, P. (1980) *Les options en éducation*. Québec : ministère de l'Education.
- Bézivin J. (2003) : La transformation de modèles. INRIA-ATLAS & Université de Nantes, 2003. Dans: *École d'Été d'Informatique cours #6*, CEA EDF INRIA.
- Bloom, B. S. (1968). Learning for mastery. *Evaluation Comment* (UCLA-CSIEP), 1(2), 1-12.
- Bloom, B.S. (1979). *Caractéristiques individuelles et apprentissage scolaire*, Bruxelles : Labor.
- Boot, E. W., J. Nelson, et al. (2007). Stratification, Elaboration, and Formalization of Design Documents: Effects on the Production of Instructional Materials. *British Journal of Educational Technology* 38(5): 917-933.
- Botturi, L. (2003). E2ML: Educational environment modelling language. Thèse de doctorat, Université de Lugano, Italie.
- Botturi, L., Cantoni, L., Lepori, B. et Tardini, S. (2006). Fast prototyping as a communication catalyst for e-learning design. In M. Bullen et D. Janes (dir.), *Making the transition to e-learning: Strategies and issues* (p. 266-283). Hershey, PA : Idea Group.
- Botturi, L. (2008). E2ML: A tool for sketching instructional designs. In L. Botturi et G. Stubbs (Eds.), *Handbook of Visual Languages for Instructional Design: Theories and Practice* (pp. 122-132). Hershey, PA: Informing Science Reference.
- Boudali, F., Balla, A. & Amrouche, H. (2009). Pertinence des Résultats de Recherche de Services Web E-Learning. In A. Amine, O. A. Mohamed & Z. Elberichi (eds.), *CIIA*: CEUR-WS.org.
- Bourgeois, E. et Nizet, J. (1999) Interactions sociales et apprentissage. In *Apprentissage et formation des adultes. Education et Formation : formation permanente éducation des adultes*. PUF - France (p.155-200)
- Bourguin, G., & Derycke, A. (2005). Systèmes interactifs en co-évolution réflexions sur les apports de la théorie de l'activité au support des pratiques collectives distribuées. *Revue d'Interaction Homme Machine*(vol. 6, n°1), 1-31.
- Brasher, A., Conole, G., Cross, S., Weller, M., Clark, P., & White, J. (2008). CompendiumLD – a tool for effective, efficient and creative learning design. In: *Proceedings of the 2008 European LAMS Conference: Practical Benefits of Learning Design*, 25-27 June 2008, Cadiz, Spain.
- Brown, A.L., & Campione, J.C. (1995). Concevoir une communauté de jeunes élèves. Leçons théoriques et pratiques. *Revue française de pédagogie*, n°11, 1995, pp. 11-33.

BIBIOGRAPHIE

- Brown, A.L., & Campione, J.C. (1996). Psychological theory and the design of innovative learning environments: On procedures, principles, and systems. In L. Schauble & R. Glaser (Eds.), *Innovations in learning: New environments for education* (pp. 289–325). Mahwah, NJ: Erlbaum.
- Bruillard, E. (1997). *Les machines à enseigner*. Paris : Éditions Hermès.
- Bruner, J. S. (1966). *Toward a theory of instruction*. Cambridge, Mass.: Belkapp Press.
- Bruner, J. S. (1991). Car la culture donne forme à l'esprit. De la révolution cognitive à la psychologie culturelle. Paris: Eshel
- Bunt, A, Conati, C, Huggett, M., Muldner, K. (2001). On improving the effectiveness of Open learning Environments through tailored support for Exploration. *Proceedings AIED*. 2001
- Burnett, M. (mise à jour 2009). 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, M. (2008). poEML: A separation of concerns proposal to instructional design. In L. Botturi et G. Stubbs (Eds.), *Handbook of Visual Languages for Instructional Design: Theories and Practice* (pp. 185-209). Hershey, PA: Informing Science Reference.
- Chinosi M. & Trombetta A. (2012). BPMN: An introduction to the standard. *Computer Standards & Interfaces* [En ligne]. 2012. Vol. 34, p. 124–134.
- Choquet C. (2007). Ingénierie et réingénierie des EIAH : l'approche REDiM. Habilitation à diriger des recherches en informatique, 179 p. Laboratoire d'Informatique de l'Université du Maine.
- Caron P.A. (2007). Bricoles: une approche dispositive des applications Web 2.0 utilisables pour enseigner. In: *Actes de la conférence EIAH 2007*, 27-29 Juin 2007, Lausanne (Suisse), p. 137-142.
- Compte C. (2004). Le renouvellement de l'accès aux savoirs. In I. Saleh (dir.), *Enseignement ouvert et à distance, épistémologie et usages* (p. 52-72). Paris : Lavoisier/ Hermès Sciences.
- 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
- Crahay, M. (1999). Psychologie de l'éducation. Paris : PUF, 1er cycle
- Crozat, S., et Trigano, P. (2002). Structuration et scénarisation de documents pédagogiques numériques dans une logique de massification. *Sciences et Techniques Educatives*. Récupéré le 3 octobre 2013 de <http://halshs.archives-ouvertes.fr/docs/00/00/18/79/PDF/ste.pdf>
- Dalziel, J. (2003) Implementing learning design: The learning activity management system (LAMS). *ASCILITE*. <http://www.melcoe.mq.edu.au/documents/ASCILITE2003%20Dalziel%20Final.pdf>
- Dede, C., Salzman, M., Loftin, B., & Sprague, D. (1999). Multisensory immersion as a modeling environment for learning complex scientific concepts. In W. Feurzeig and N. Roberts, (Eds.), *Computer modeling and simulation in science education*, New York: Springer-Verlag, 282-319
- Delacour, G. (2008). E-learning : le devenir de la conception pour le formateur, *ilearning Forum 2008*, Paris, France.
- Derntl, M. & Motschnig-Pitrik, R. (2008). coUML: A visual language for modeling cooperative environments. In L. Botturi et G. Stubbs (Eds.), *Handbook of Visual Languages for Instructional Design: Theories and Practice* (pp. 155-184). Hershey, PA: Informing Science Reference.
- Dessus, P. (2006). Quelles idées sur l'enseignement nous révèlent les modèles d'instructional design ? [version électronique] *Revue suisse des sciences de l'éducation*, 28(1), 137-157. Récupéré de la page de l'auteur : [http://web.upmf-grenoble.fr/sciedu/pdессус/rsse06.pdf](http://web.upmf-grenoble.fr/sciedu/pdессus/rsse06.pdf)

BIBIOGRAPHIE

- Dillenbourg, P., & Hong, F. (2008). The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning*. 3 (1), pp. 5-23
- Dillenbourg, P. & Jermann, P. (2002). Internet au service de l'innovation. In R. Guir (Ed.), *Pratiquer les TICE* (pp. 179-196). Bruxelles : De Boeck.
- De Vries, F. et al (2005), Future developments of IMS Learning Design tooling, Pre-Discussion Paper, UNFOLD project, Valkenburg (The Netherlands), February 16th-18th 2005
- Doise. W. (1993). *Logique sociales dans le raisonnement*. Neuchâtel, Delachaux et Niestlé.
- Doise, W. & Mugny G. (1981). *Le Développement Social de l'intelligence*. Paris : InterEditions.
- Doise, W. & Mugny, G. (1997). *Psychologie sociale et développement cognitif*. Paris: Ed. Armand Colin.
- Doolittle, P. E., (1999). *Constructivism and online education*. Virginia : Polytechnic Institute & State University.
- Ernst, C. (2006). Cadre conceptuel pour former à la scénarisation d'un enseignement médiatisé. *Revue de l'éducation à distance*, 21(2), 58-77.
- Engeström, Y. (1987). *Learning by expanding: An activity-theoretical approach to developmental research*. Helsinki: Orienta-Konsultit.
- Faure D., Lejeune A. (2005). GenScen' Un éditeur graphique pour l'enseignant scénariste, *EIAH 2005*, Montpellier, mai 2005, pp 431-436, 2005.
- Favre J-M, Estublier J., Blay-Fornarino M., 2006, L'ingénierie dirigée par les modèles. Au-delà du MDA, Traité IC2, Série Informatique et Systèmes d'Information, Editions Lavoisier, 2006
- Ferri, F. (2008). *Visual Languages for Interactive Computing: Definitions and Formalizations* (pp. 1-532). Hershey, PA: IGI Global. doi:10.4018/978-1-59904-534-4
- 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
- Figl, K., Derntl, M., Rodriguez, M.C., Botturi, L. (2010). Cognitive Effectiveness of Visual Instructional Design Languages. *Journal of Visual Languages and Computing*, 21 (6), 359–373.
- Gagné, E. D. (1985). *The cognitive psychology of school learning*. Boston : Little brown & Company.
- Gagné, R. M. (1974). *Essentials of learning for instruction*. (2nd ed.). Hinsdale, IL: The Dryden Press.
- Gagné, R. (1985). *The Conditions of Learning* (4th.). New York: Holt, Rinehart & Winston.
- Gagné, R. & Driscoll, M. (1988). *Essentials of Learning for Instruction* (2nd Ed.). Englewood Cliffs, NJ: Prentice-Hall.
- Gagné, R., Briggs, L. & Wager, W. (1992). *Principles of Instructional Design* (4th Ed.). Fort Worth, TX: HBJ College Publishers.
- Gartner Inc. (2005) *Business Process Management Suites Enhance the Control and Management of Business Processes*.
- Gautier D. (2004). DAML-S:interactions, critique et évaluation. Institut d'informatique des FUNDP. Namur, Belgique.
- Gilly, M. (1989). Les représentations sociales dans le champ éducatif. In D. Jodelet (Ed.), *Les représentations sociales* (pp. 363-385). Paris: Delachaux et Niestlé.
- Gilly, M. (1995). Interactions sociales et constructions cognitives. In Gaonac'h D. & Golder C., *Profession enseignante : Manuel de psychologie pour l'enseignement*, Paris: Hachette. 136-154.

BIBIOGRAPHIE

- Koper, R. (2002). Modeling units of study from a pedagogical perspective – The pedagogical metamodel behind EML. Récupéré en décembre 2013 depuis : <http://citeserx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.445&rep=rep1&type=pdf>
- Helic, D. (2007). Managing Collaborative Learning Processes in eLearning Applications. *Proceedings of the ITI 2007 29th International Conference on information technology interfaces*, Croatia, 2007, pp.345-350.
- Henri, F., Compte, C. & Charlier, B. (2007). La scénarisation pédagogique dans tous ses débats. PROFETIC, *Revue internationale des technologies en pédagogie universitaire*. <http://www.ritpu.org/spip.php?article132>
- 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
- Jochems, W. M. G., J. J. G. van Merrienboer, et al., Eds. (2003). *Integrated E-Learning: Implications for Pedagogy, Technology and Organization*. London, Routledge Farmer.
- Jonassen, D. H. (1994). Technology as cognitive tools: Learners as designers. *IT Forums Paper*. Récupéré en décembre 2013 depuis <http://itech1.coe.uga.edu/itforum/paper1/paper1.html>.
- Kleppe A., Warmer J., Bast, W (2003). *MDA EXPLAINED the Model Driven Architecture: Practice and Promise*. Addison- Wesley (2003).
- Kobbe, L. (2006). Framework on multiple goal dimensions for computer-supported scripts, Kaleidoscope, D21.2.1 (Final). [Retrieved july 12 2007 from <http://telearn.noe-kaleidoscope.org/browse/publications/keyword/?keyword=pedagogical%20scenarii>]
- Koper R., (2000). From change to renewal: Educational technology foundations of electronic learning environments. Open University of the Netherlands <http://dspace.learning networks.org/retrieve/37/kop> dernière consultation, octobre 2013.
- Marriott, K. et Meyer, B. (1998). *Visual language theory*. New York : Springer-Verlag.
- Martel, C., Lejeune, A., Ferraris, C. et Vignollet, L. (2007). Scénariser les 4 piliers de la pédagogie. Dans T. Nodenot, J. Wallet et E. Fernandes (dir.), *Actes de la conférence Environnement informatique pour l'apprentissage humain (EIAH 2007)* (p. 269-274).
- Martel, C., Vignollet, L., Ferraris, C., & Durand, G., (2006). Ldl : a language to model collaborative learning activities. In *EDMEDIA 2006, World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 838– 844, 2006.
- Matusevich, M. N. (1995). School reform: What role can technology play in a constructivist setting? <http://pixel.cs.vt.edu/edu/fis/techcons.html>
- Laborde, C. & Capponi, B. (1994). Cabri-géomètre constituant d'un milieu pour l'apprentissage de la notion de figure géométrique. In BALACHEFF, N. & VIVET, M. (Eds.), *Didactique et intelligence artificielle* (pp. 165-210 ed.). Grenoble : La Pensée sauvage.
- Legendre, R. (1993). Dictionnaire actuel de l'éducation (2e édition). Montréal/Paris: Guérin/ESKA.
- Lejeune, A. (2004). IMS Learning Design, dans *Distances et savoirs* 4/2004 (Vol. 2), p. 409-450. URL : www.cairn.info/revue-distances-et-savoirs-2004-4-page-409.htm
- Lévi-Strauss, C. (1962). *La pensée sauvage*. Paris, Plon.
- Lindsay, P. H. & Norman D. A. (1980). *Traitement de l'information et comportement humain*. Montréal: Études vivantes.
- Lipponen, L. (2002), Exploring foundations for computer-supported collaborative learning, Proceedings of CSCL 2002.

BIBIOGRAPHIE

- Long P. & Siemens G. (2011) Penetrating the fog: analytics in learning and education. *EDUCAUSE Review* 46, 31–40. Depuis <http://www.educause.edu/ero/article/penetrating-fog-analytics-learning-and-education>
- Martel, C., Vignollet, L., Ferraris, C., David, J.P., Lejeune, A. (2006). Modeling collaborative learning activities on e-learning platforms, *6th IEEE ICALT Proceedings*, Kerkrade, 2006, p. 707-709
- Morales, L., Castillo, L. A., Fernández-Olivares, J., and González-Ferrer, A. (2008). Automatic generation of user adapted learning designs: An ai-planning proposal. In Nejdl, W., Kay, J., Pu, P., and Herder, E., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, 5th International Conference, AH 2008*, Hannover, Germany, July 29 - August 1, 2008. Proceedings, volume 5149 of *Lecture Notes in Computer Science*, pages 324–328. Springer.
- Nardi, B. (1996). Concepts of Cognition and Consciousness: Four Voices. Invited paper, *Australian Journal of Information Systems*, 4, 1, 64-79. September. Reprinted in *ACM Journal of Computer Documentation*, February, 1998.
- Nodenot T., Scénarisation pédagogique et modèles conceptuels d'un EIAH : Que peuvent apporter les langages visuels ?. *International Journal of Technologies in Higher Education (IJTHE)* 4, 2, p. 85-102, (2007)
- Nodenot, T., Laforcade, P. et Le Pallec, X. (2008). Visual design of coherent technology-enhanced learning systems: A few lessons learned from CPM language. In L. Botturi et G. Stubbs (Eds.), *Handbook of Visual Languages for Instructional Design: Theories and Practice* (pp. 254-280). Hershey, PA: Informing Science Reference.
- Occello, A. (2002). Composants: Vers une adaptation dynamique cohérente. *DEA d'Informatique*. Université de Nice Sophia Antipolis. Rapport de stage présenté en juillet 2002.
- OMG-BPMN 2.0, <http://www.omg.org/spec/BPMN/2.0.1/PDF>, 2013.
- Paquette, G. (2004). Educational Modeling Languages, from an Instructional Engineering Perspective. Dans R. McGreal (dir.), *Online education using learning objects* (p. 331-346). London : Routledge/Palmer.
- Paquette, G. (2005). *L'ingénierie pédagogique, pour construire l'apprentissage en réseau*. Sainte-Foy, Presse de l'Université du Québec.
- Pavlov, I.P. (1901). *Le travail des glandes digestives*. Paris Masson et Cie.
- Pavlov, I. P. (1927). *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Translated and Edited by G. V. Anrep. London: Oxford University Press.
- Piaget, J. (1936). *La naissance de l'intelligence chez l'enfant*. Genève: Delachaux et Niestlé.
- Piaget, J. (1970). *L'épistémologie génétique*. Paris, PUF: coll. Que sais-je?
- Piaget, J. (1975). *L'équilibration des structures cognitives*. Paris, PUF.
- Pernin, J.-P. (2003). Objets pédagogiques : unités d'apprentissage, activités ou ressources ?. *Revue "Sciences et Techniques Educatives"*. Hors-série 2003, Ressources numériques, XML et éducation, pp 179-210, avril 2003, éditions Hermès
- Pernin, J.-P. (2004). LOM, SCORM et IMS-Learning Design : ressources, activités et Scénarios. Récupéré le 10 mars 2011 du site de l'École nationale supérieure des sciences de l'information et des bibliothèques : <http://www.enssib.fr/bibliotheque-numerique/document-1810>
- Pernin J.-P. & Lejeune A. (2004). Modèles pour la réutilisation de scénarios d'apprentissage. *Actes du colloque TICE Méditerranée*, p.48, Nice, novembre 2004.
- Perret-Clermont, A-N. (1979). *La Construction de l'Intelligence dans l'Interaction Sociale*. Genève , Ed. Peter Lang.
- Perrenoud, P. (2003). Êtes-vous pour ou contre la gravitation universelle. *Résonances*, novembre 2003, n° 3, p. 7-9

BIBIOGRAPHIE

- Rawlings, A., Van Rosmalen, P., Koper, R. Rodriguez-Artacho, M., Lefrere, P. (2002). Survey of Educational Modelling Languages (EMLs). *Learning Technologies Workshop Version 1, CEN/ISSS WS/LT, 19. Septembre 2002.*
- Recker, J. (2010). Opportunities and Constraints: The Current Struggle with BPMN. *Business Process Management Journal*, 2010. 16(1): p. 181-201.
- Recker, J., Safrudin, N., & Rosemann, M. (2010). How Novices Model Business Processes. In Hull, R., Mendling, J., & Tai, S., (Eds.) *Business Process Management - BPM 2010*, Springer, Hoboken, New Jersey, pp. 29-44. <http://eprints.qut.edu.au/33203/>
- Reigeluth, C. M. (1999). What is instructional-design theory and how is it changing ? In C. M. Reigeluth (Ed.), *Instructional-design theories and models: A new paradigm of instructional theory* (Vol. II, pp. 5-29). Mahwah, NJ: Lawrence Erlbaum Associates.
- Reichert, R., & Hartmann, W. (2004). On the Learning in E-Learning. In ED-MEDIA 2004 : *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, pp.1 CD-Rom. Norfolk, Va.: AACE.
- Richelle, M. (1977). *B. F. Skinner: Ou, Le péril behavioriste*. Bruxelles: P. Mardaga.
- Schneider, D. et al. (2003). Conception et implémentation de scénarios pédagogiques riches avec des portails communautaires. *Colloque de Guéret 4-6 Juin 2003*
- Schneuwly, B. & Bronckart, J-P. (1985). *Vygotski aujourd'hui*. Neuchâtel : Delachaux & Niestlé.
- Siemens, G. (2002). Instructional design in elearning. Récupéré le 20 octobre 2013 depuis <http://www.elearnspace.org/Articles/InstructionalDesign.htm>
- Silver, B., (2009) *BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0*. Vol. New York. 2009: Cody-Cassidy Press.
- Singh, H. (2003). *Building effective blended learning programs*. Educational Technology, 51-54.
- Skinner B. F. (1957). *Verbal behavior*. Cambridge, MA: Prentice Hall.
- Skinner B. F. (1968). *La révolution scientifique de l'enseignement (The technology of teaching)*, 3^e édit., 1995, Bruxelles: P. Mardaga.
- Stahl G. (2002) Contributions to a theoretical framework for CSCL in G. Stahl (Ed.), (2002) *Computer support for collaborative learning: foundations for a CSCL community*, (Cscl 2002 Proceedings), Mahwah, NJ: Lawrence Erlbaum Associates.
- Tardif, J. (1992). Pour un enseignement stratégique: l'apport de la psychologie cognitive. Montréal: Éditions Logiques.
- Tattersall, C., Koper, R. (2003). EML and IMS Learning Design: from LO to LA. *Educational Technology Expertise Centre, The Open University of the Netherlands* http://dspace.learningnetworks.org/retrieve/186/_LTSN+Presentation+-+EML+and+LD+20030238.pdf. Dernier accès, novembre 2013.
- Taurisson, A. (2005). La pédagogie de l'activité, un nouveau paradigme ? De l'intention à la réalisation pédagogique. *Thèse de doctorat en sciences de l'éducation*, Université de Lyon II.
- Samba, D. (2011). SPEM4MDE : Un métamodèle et un environnement pour la modélisation et la mise en œuvre assistée de processus IDM. *Thèse d'université*, Université Toulouse II, Octobre 2011.
- Sampson, D., Karampiperis, P. et Zervas, P. (2005). ASK-LDT: A Webbased learning scenarios authoring environment based on IMS learning design. *Advanced Technology for Learning*, 2(4), 207-215
- Tchounikine, P. (2004). Platon-1 quelques dimensions pour l'analyse des travaux de recherche en conception d'EIAH. <http://halshs.archives-ouvertes.fr/> Référence HAL: csqd-00002999, version 1.
- Thorndike, E. (1922). *The Psychology of Arithmetic*. New York : Macmillan.

BIBIOGRAPHIE

- Thorndike, E. (1932). *The Fundamentals of Learning*. New York : Teachers College Press.
- Van Merriënboer, J. J. G., & Kirschner, P. A. (2001). Three worlds of instructional design: State of the art and future directions. *Instructional Science*, 29, 429-441.
- Vantroys, T. & Peter, Y. (2005). COW, un service de support d'exécution de scénarios pédagogiques, STICEF, Vol. 12, 2005.
- Vygotski, L.S. (1962). *Thought and Language*. Cambridge, MA: MIT Press.
- Vygotski, L.S. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.
- Watson, J. B. (1913). Psychology as the behaviorist views it. *Psychological Review*. 20, 158-177.
- WfMC, (1999). *WFMC-TC-1011 – terminology Glossary*. http://www.aiim.org/wfmc/standards/_conformance.htm accédé en décembre 2013

ANNEXES

ANNEXES

ANNEXE 1 - HISTORIQUE DES COURANTS DE L'APPRENTISSAGE

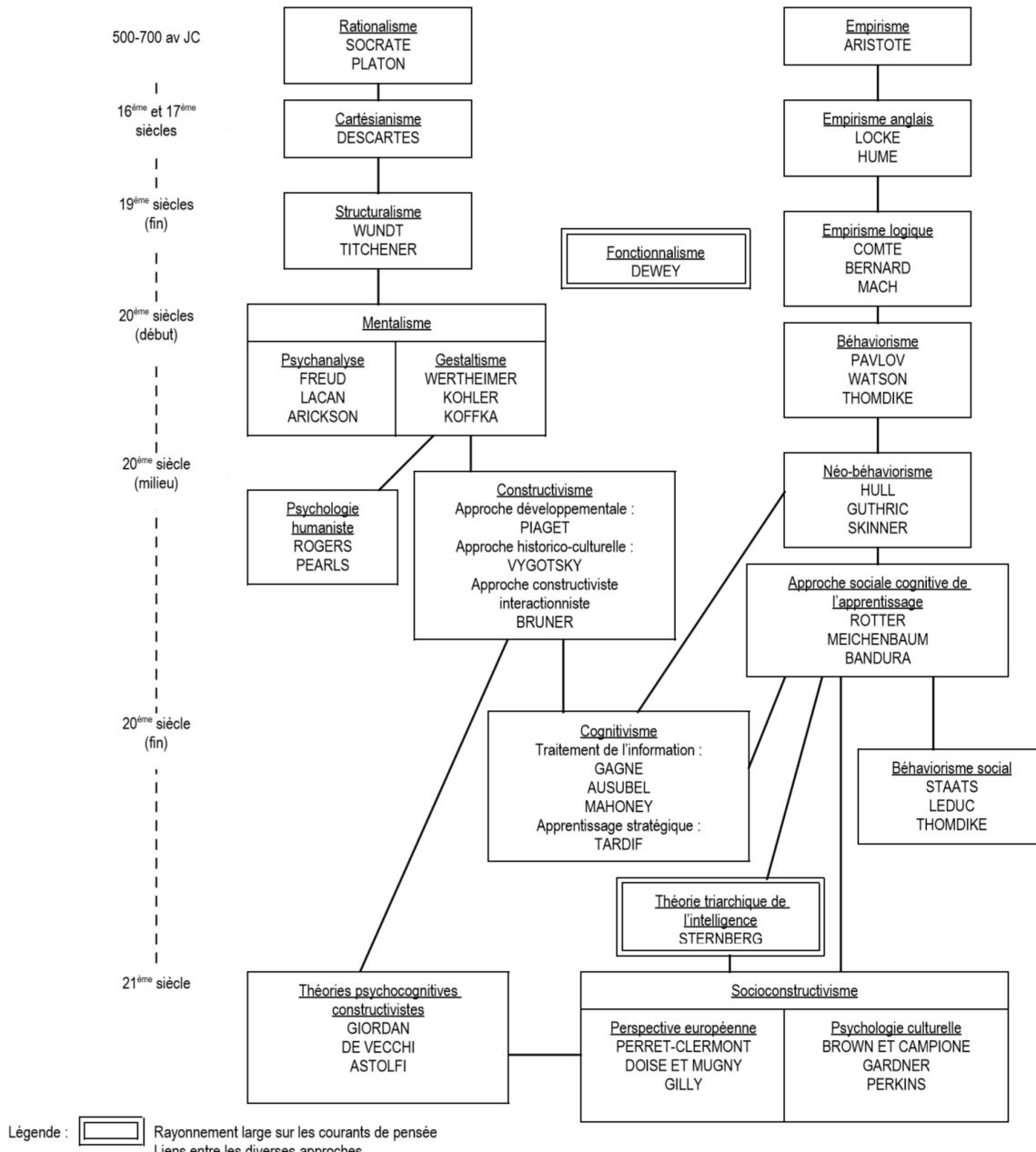


Figure 49: Minier, P., (2003). Anrage historique et développement des courants de pensée de l'apprentissage.
<http://wwwens.uqac.ca/~pminier/act1/graph1.htm>

ANNEXE 2 – COMPLÉMENT IMS LD

A2-1 Définition d'une unité d'apprentissage et modèle structurel.

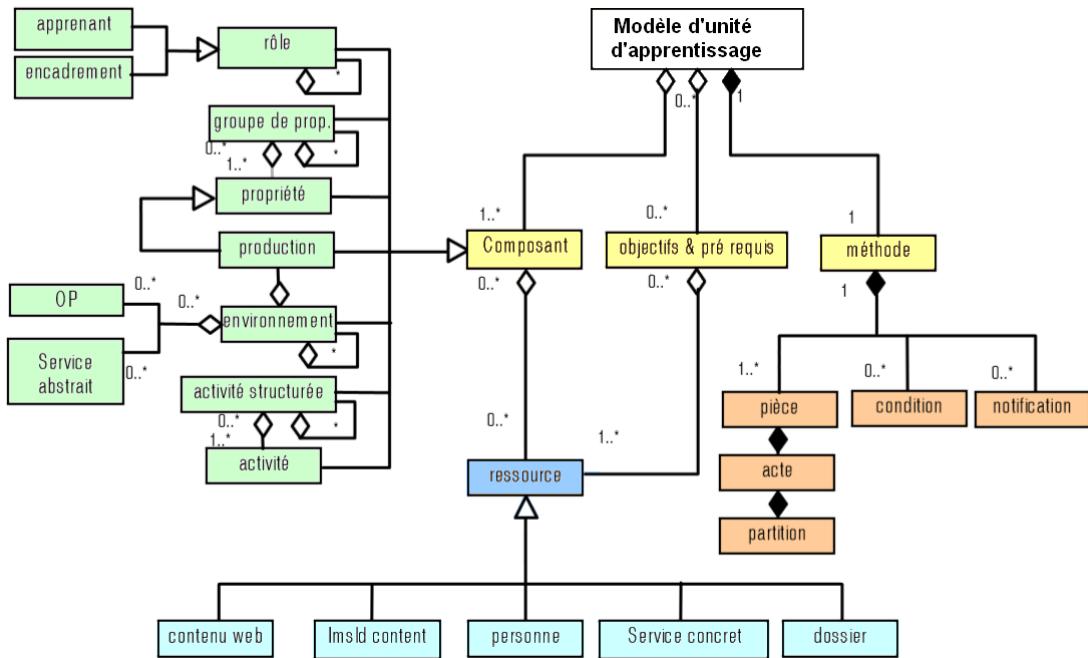
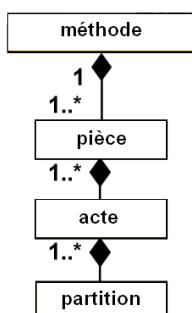


Figure 50: Modèle structurel IMS-LD d'une unité d'apprentissage. (Adapté depuis IMS LD information Model)

Dans IMS-LD, un modèle d'unité d'apprentissage est construit à partir de trois éléments, (1) les *composants*, (2) les *objectifs et prérequis* et (3) la *méthode*. Nous les avons colorés en jaune sur le diagramme UML de la figure 50.

Composants : D'après la documentation IMS, les composants permettent de décrire dans un formalisme XML les entités nécessaires à la mise en place d'une unité d'apprentissage (En vert, figure 50). On peut ainsi décrire les rôles impliqués, les différentes activités, la description des propriétés d'un rôle ou d'une personne, les services manipulé etc.

Objectifs et prérequis : les objectifs d'apprentissage décrivent les compétences ou les connaissances devant être acquises à l'issue de l'unité d'apprentissage. Les prérequis spécifient l'ensemble des conditions préalables à son suivi. Ces indications peuvent être rédigées dans un format textuel, un format web (xhtml) ou encore via l'inclusion de la spécification IMS RDCEAO (Reusable Definition of Competency or Educational Objective).



Méthode : Si comme nous l'avons précédemment évoqué, l'origine du terme « scénarisation pédagogique » est issue du monde des arts audiovisuels, IMS-LD nous rappelle cette origine en étant basé sur une métaphore théâtrale de l'enseignement. Le déroulement d'une unité d'apprentissage repose sur l'élément *méthode* qui en décrit les différents déploiements sous forme de pièces (mises en scène). Une pièce est composée d'actes exécutés en séquences. Enfin, les actes sont formés de partitions qui associent un rôle à une activité effectuée dans un environnement (décor).

A2-2 Niveaux de conception et modèle relationnel :

Il y a 3 niveaux de conception dans IMS-LD respectivement définis par les lettres A, B et C.

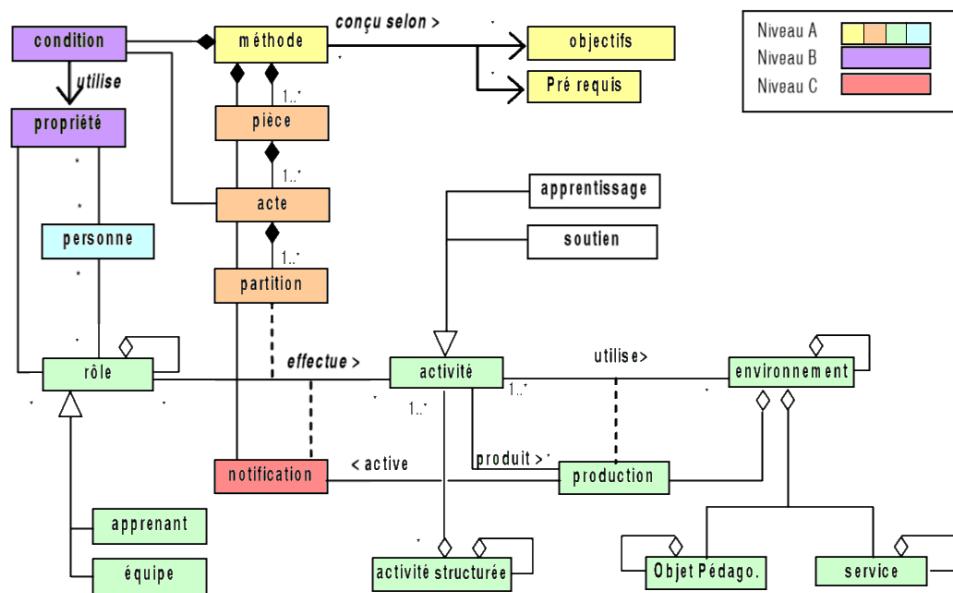


Figure 51: Modèle relationnel d'IMS-LD (Adapté depuis IMS LD information Model)

Le niveau A est celui de base. Ce niveau permet la spécification des activités pédagogiques ordonnées dans le temps et effectuées par chacun des rôles définis dans le contexte d'un milieu composé d'*objets d'apprentissage* et de *services*. L'unité d'apprentissage se déroule selon la *méthode* (orange) qui est une séquence d'éléments permettant de définir la dynamique du processus d'apprentissage. Une méthode part des conditions initiales (pré requis) qui doivent être satisfaites et qui a comme but d'atteindre certains objets d'apprentissage (en jaune). La méthode effectue ensuite le déroulement des activités en fonction des rôles (en vert). Ces activités sont en relation avec un environnement composé de ressources pédagogiques et de services.

Le niveau B intègre le niveau A mais ajoute des propriétés et des conditions (en violet) permettant une représentation du processus d'apprentissage plus avancée. Les propriétés sont utilisées pour stocker les informations sur une personne ou sur un groupe de personnes. Les conditions permettent de décider de l'évolution d'un scénario pédagogique à un moment donné. Par l'évaluation d'une expression on peut décider, en fonction de son résultat, quel parcours suivra le scénario.

Le niveau C intègre le niveau B en y ajoutant les *notifications*. Ce niveau C permet la transmission des messages d'un rôle ou l'ajout de nouvelles activités associées à un rôle, qui seraient des conséquences de l'apparition des événements pendant le processus d'apprentissage.

ANNEXE 3 – PoEML PERSPECTIVES.

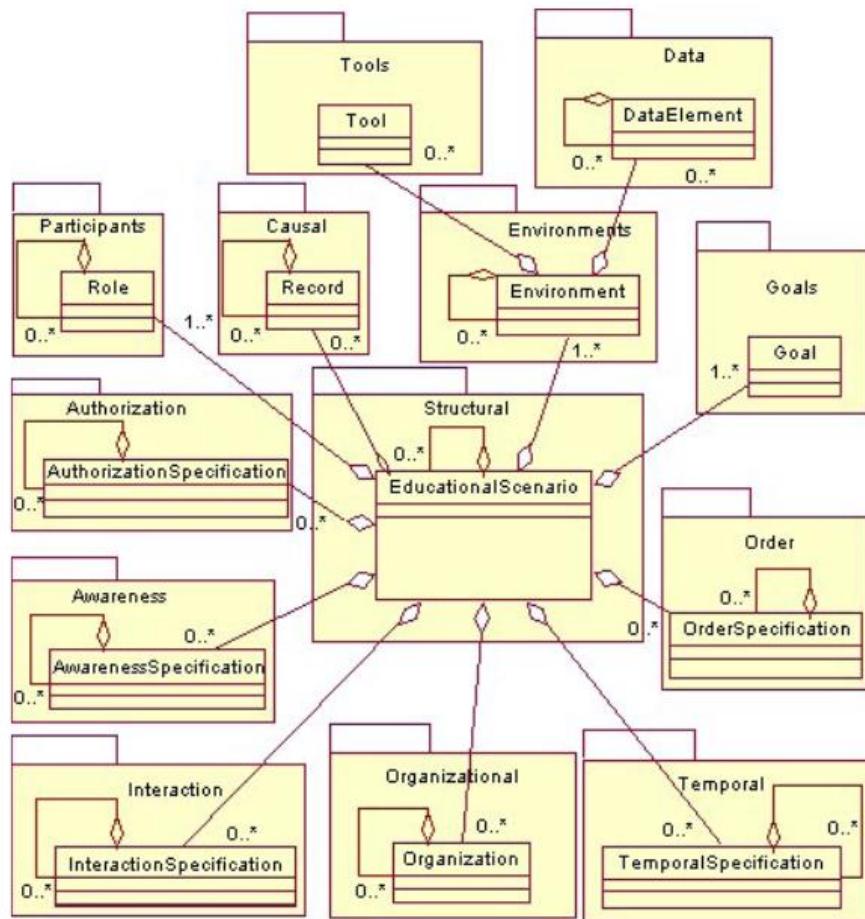


Figure 52: Modèle UML des 13 perspectives de poEML et leurs relations

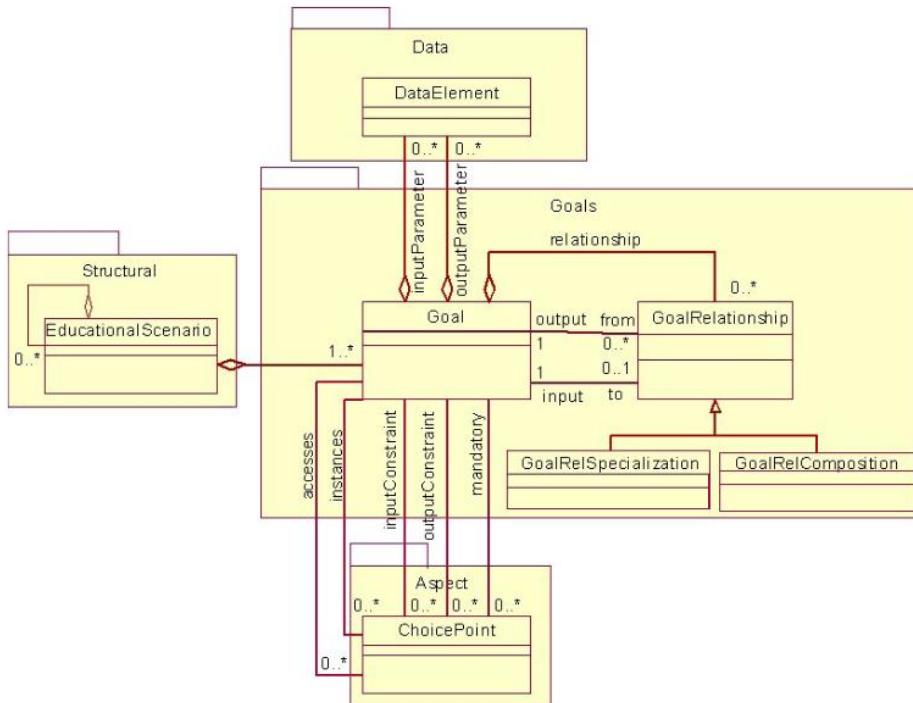


Figure 53: Modèle UML des éléments principaux de la perspective "Goal"

ANNEXE 4 - COMPLÉMENT AU CHAPITRE 4 - BPMN 2.0

Nous fournissons ici un complément d'information sur la structure du langage BPMN, son fonctionnement, les éléments qui le compose et les types de diagrammes.

A4-1. Structure du langage BPMN

A4-1.1 Sous-classes de conformité

BPMN est constitué d'un ensemble d'éléments de modélisation. Ces éléments sont constitués d'un symbole (objet graphique sur un diagramme) et d'une liste d'attributs (invisible sur un diagramme). Ces éléments augmentent en nombres et en complexité d'attribut en fonction du niveau de conformité à la norme 2.0. La spécification définit trois sous-classes de conformité (conformance sub-classes) :

- *Descriptive* : Un peu plus d'une vingtaine d'éléments se basant sur les éléments et attributs nécessaires à la description d'un processus de haut niveaux.
- *Analytique* : Contient tous les éléments précédents et en ajoutent de nouveaux. Contient une cinquantaine d'éléments et ajoutent quelques attributs aux précédents.
- *Exécutable* : elle contient tous les symboles BPMN 2.0 et un grand nombre d'attributs d'éléments de processus pour concevoir des modèles et leur exécution automatique.

À titre d'illustration nous comparons ci-dessous le même élément selon la classe de conformité descriptive (niveau 1) et exécutable (niveau 3).

Name	Symbol	Attributs classe descriptive	Attributs classe exécutable
Tâche utilisateur (<i>userTask</i>)		id, name	Id, name, renderings, implementation, resources, ioSpecification, dataInputAssociations, dataOutputAssociations, loopCharacteristics, boundaryEventRefs.

Tableau 3: Comparaison d'un élément BPMN selon sous-classe de conformité

Les outils de modélisation les plus simples utiliseront le niveau de conformité 1 tandis que les systèmes complets de gestion de procédure métier (BPMS) utiliseront le niveau 3.

A4-1.2 Meta-modèles

La spécification des versions précédentes de BPMN (jusqu'à BPMN 1.2) ne contenait que des descriptions verbales des éléments de notation graphique et des règles de modélisation. La spécification de BPMN 2.0 quant à elle fournit des diagrammes de classes UML complets montrant graphiquement les caractéristiques des différents construits BPMN et leurs relations. Ces diagrammes de classes donnent une représentation statique d'un système constitué d'objets logiques assimilables à des classes, en lien les uns avec les autres. Ils sont composés de packages, de classes, d'attributs et d'opérations de classes, d'associations, de compositions, d'agrégation et de généralisation entre classes. Plus d'une cinquantaine de diagramme sont définis par la norme et formalisent une sémantique d'exécution pour BPMN. La figure 54 est un exemple de diagramme de classes. Il représente la classe « Definitions », un des objets les plus périphériques définissant le champ d'application et l'espace de noms pour tous les éléments BPMN. Il permet notamment l'échange de fichier BPMN.

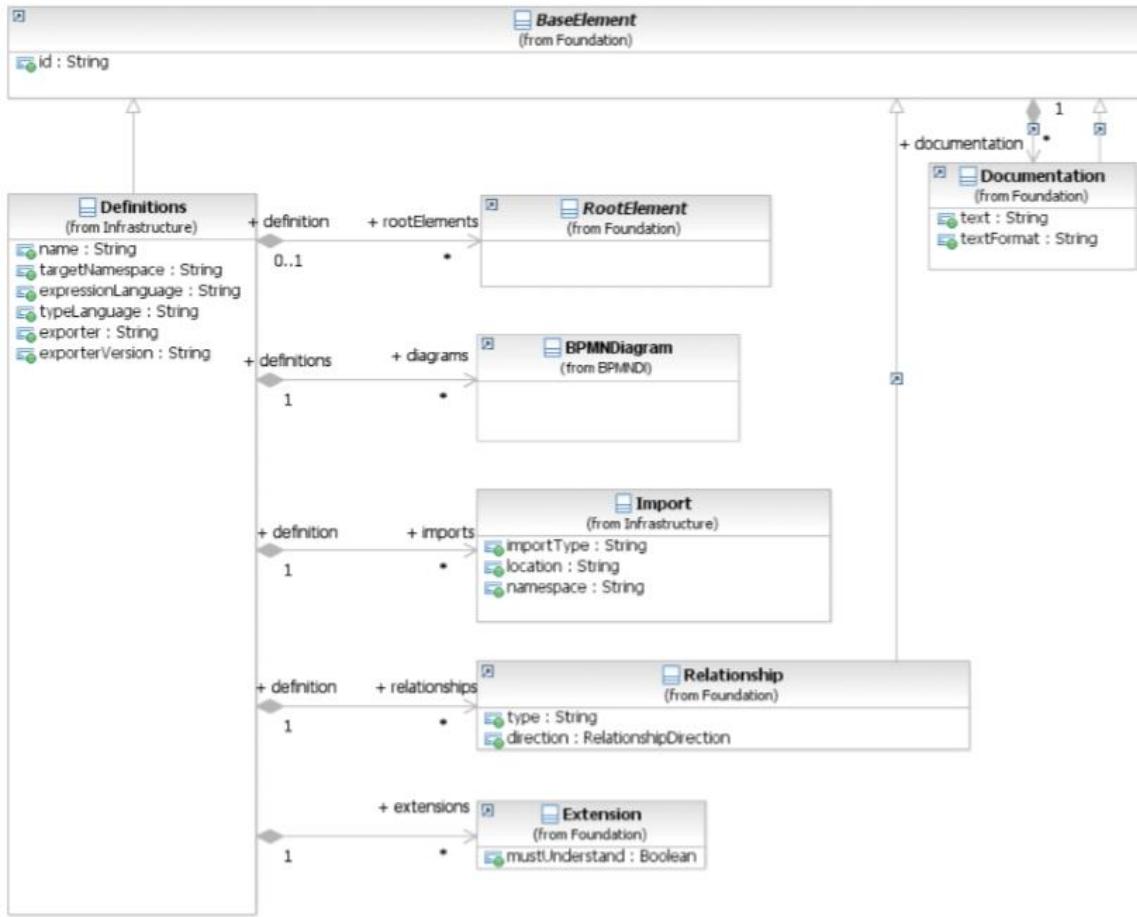


Figure 54: Diagramme de classe "Definitions" (OMG, 2013, p.55)

Concernant la structure générale du langage la notation est organisée selon un concept de layers (couches) extensibles. Chaque couche est construite en se basant sur la couche précédente (Figure 55). Cette structure permettra au langage d'évoluer en ajoutant des couches sans remettre en cause le cours du langage.

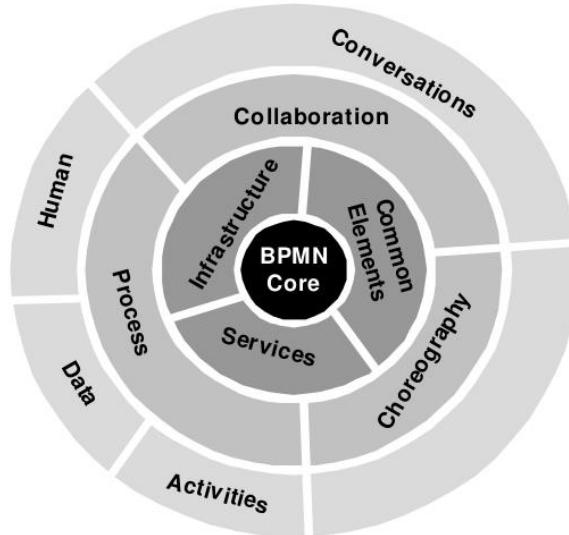


Figure 55: BPMN 2.0 structure en couche extensible

A4-2. Les types de symboles BPMN

A4-2.1 Élément de flux

Nom	Symbol	Brève description
Tâche (<i>task</i>)		Réalisation d'une action par un utilisateur ou un système informatique
Porte (<i>Gateway</i>)		Gestion des embranchements du flux d'activité. Associer, dissocier, dédoubler les chemins.
Évènement (<i>event</i>)		Quelque chose se produisant au cours du processus et ayant un impact sur lui (ex: Départ, arrêt, pause.)

Tableau 4: Élément de flux

Les éléments présentés tableau 3 constituent la base de la modélisation avec BPMN. Chacun de ces symboles est ensuite déclinés en certains nombres de variantes précisant leur sens. À titre d'exemple, ci-dessous des détails additionnels permettent de différencier une *tâche humaine*, d'une *tâche informatique* ou différencier une *porte inclusive* (« AND ») d'une *porte exclusive* (« XOR ²⁶ »).



Les *événements* quant à eux sont toujours représentés de manière sphérique et sont déclinés de la même manière via des représentations graphiques additionnelles à l'intérieur du cercle. Cependant, on distingue trois sortes d'événement :



- Les *événements de départ* : ce sont des événements déclencheurs du processus. Ils sont symbolisés par un rond au trait fin. Ils peuvent se déclencher sans autre ou à réception d'un évènement particulier. Les évènements de départ reçoivent et déclenchent un processus mais n'émettent pas.
- Les *événements intermédiaires* : Permettent d'indiquer l'attente d'un type d'évènement au milieu d'un flux. Par exemple, tant que l'évènement « réception d'un message » est en attente, le flux est stoppé. Ils sont symbolisés par un rond avec un trait extérieur double. Les évènements intermédiaires peuvent recevoir ou émettre un évènement selon le cas.
- Les *événements de fin* : Ils terminent un processus et peuvent émettre une information. Ils sont symbolisés par un trait extérieur épais. Les évènements de fin émettent et terminent un processus mais ne reçoivent rien.

Un événement de type message sera représenté selon la règle précédente de la manière suivante.



²⁶ Exclusive OR : le « Ou » exclusif, bien connu en logique et programmation informatique.

A4-2.2 Éléments de connexion

Nom	Symbol	Brève description
Flux de séquence (Sequence Flow)	→	Représente le cours du processus. Permet de lier les éléments du tableau précédents.
Flux de message (Message flow)	○-----→	Représente le flux des communications. Élément lié à un message.
Association→	Aucun lien avec le cours du processus. Ce symbole sert à lier des informations additionnelles facultatives à des éléments du processus (Voir artefacts <i>tableau 6</i>).

Tableau 5: Élément de connexion

De manière simple, les événement et tâches - présentés au sous-point précédent - sont liés par des éléments de connexion, nommé flux de séquence (*tableau 4*) représentant le cours du processus, son « flux ». Les portes représentent des points de jonction où le flux se divise en chemins parallèles, conditionnels, etc. La *figure 56* illustre le principe de base de la modélisation avec BPMN.

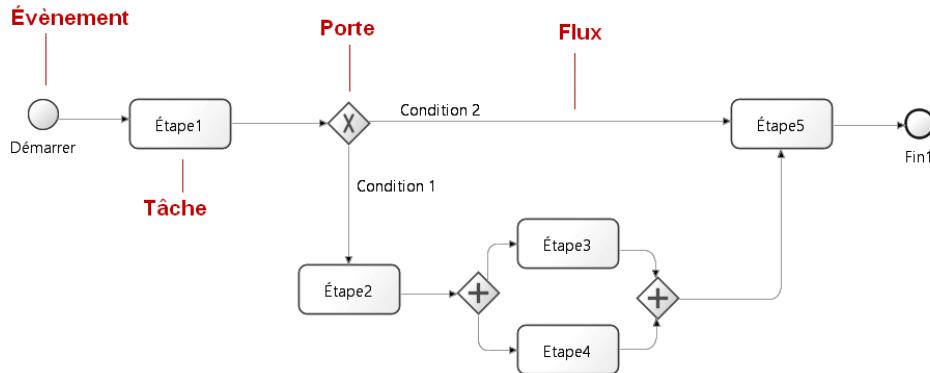


Figure 56: Illustration d'éléments simples dans un processus BPMN

Les « éléments flux de message » représentent le flux des communications. Ils transmettent de l'information et sont toujours représentés par un trait « en pointillé ». Enfin les éléments associations ne représentent aucun flux et servent simplement à lier des artefacts (voir point 3.4).

Les processus créés à l'aide des éléments de flux et de connexion peuvent ensuite être placés dans des bassins et structurés en couloirs représentant des groupes d'acteur prédéfinis grâce à deux représentations standardisées que nous présentons ci-dessous.

A4-2.3 Bassins et couloirs

Nom	Symbol	Brève description
Bassin (Pool)	Name	C'est un conteneur. Il représente les frontières d'un processus. Toutes les tâches se déroulent à l'intérieur du bassin. Seules les communications (flux de message) peuvent sortir d'un bassin vers un autre bassin.
Couloir (Lane)	Name Name	Un couloir représente un acteur, un rôle à l'intérieur d'un bassin. Le flux d'activité peut traverser les couloirs pour représenter l'enchaînement des tâches entre les différents acteurs de notre processus.

Tableau 6: Éléments BPMN « bassin » et « couloir »

Les *bassins* contiennent les processus dans leur ensemble. Ils permettent d'en définir les limites. Le flux d'activité ne peut sortir des limites d'un *bassin*. Seul les flux de message peuvent en sortir pour effectuer des communications entre des processus distincts (ex : fin d'un processus déclenche le départ d'un autre).

Afin de délimiter les étapes effectuées par différents groupes d'utilisateur BPMN permet de séparer les *bassins* en différents *coulloirs*. Il est ainsi possible de spécifier un certain nombre de rôles et de visuellement observer leur implication dans le processus. Au-delà de l'aspect visuel les *coulloirs* facilitent l'implémentation en attribuant automatiquement les tâches à des utilisateurs en fonction de leur rôle. Ce procédé fait de BPMN un langage particulièrement intéressant pour l'éducation dont les praticiens ont l'habitude de scénariser des activités en définissant des rôles. Ci-dessous un diagramme simple en BPMN illustre le fonctionnement des *bassins* et des *coulloirs*.

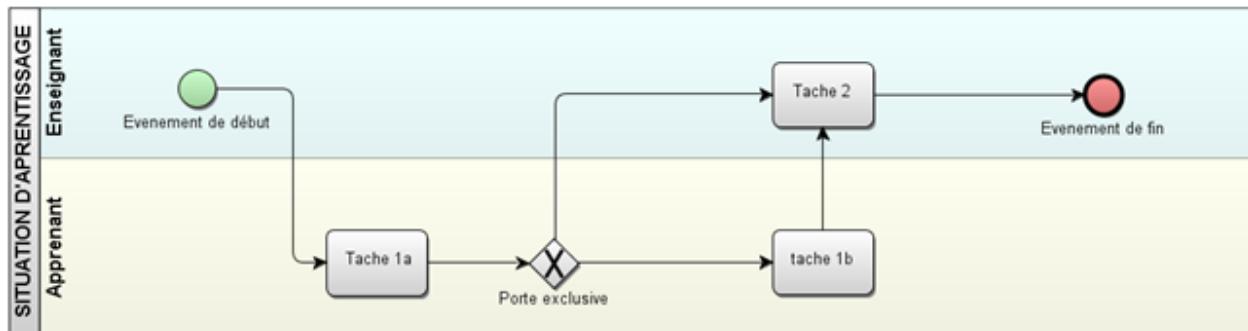


Figure 57: Illustration fonctionnement des éléments "bassin" et "couloir"

Ci-dessous un exemple avec deux bassins symbolisant deux processus différents communiquant par flux de messages. Ces diagrammes sont appelés diagrammes de collaboration (Voir point 4 pour les différentes sortes de diagrammes).

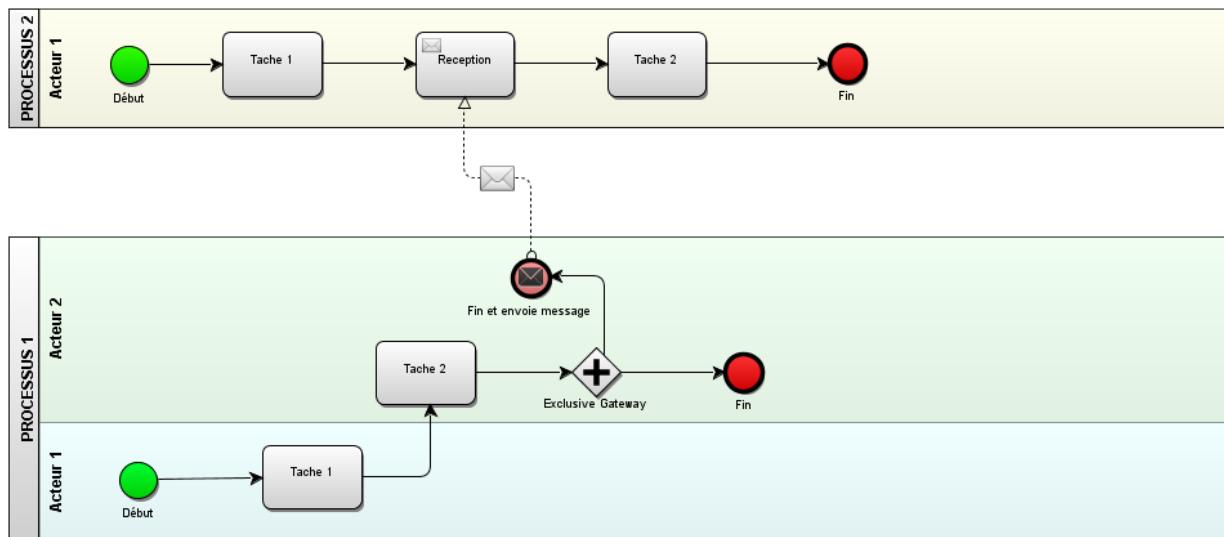


Figure 58: Illustration diagramme de collaboration

Il n'est pas obligatoire de modéliser tous les processus d'un diagramme de collaboration. Sur la figure 58 ci-dessus le processus 2 aurait pu être vide. Le flux de message se connecterait alors simplement aux limites du bassin. Le bassin sert alors à représenter une entité nommée avec laquelle notre processus communique.

La figure 58 introduit un nouvel élément, une lettre disposée sur le flux de message. Cet élément est un artefact facultatif. La spécification définit différents artefacts permettant d'apporter de l'information visuelle n'ayant aucun impact sur le cours du processus. On peut les retirer sans perturber l'activité et ils sont tous simplement ignorés par le système d'exécution de processus. Nous présentons les artefacts BPMN au sous-point suivant.

A4-2.4 Artefacts

Nom	Symbole	Brève description
Objet de Données (Data object)		Permet de représenter de manière visuelle ce dont une tâche a besoin ou ce qu'une tâche produit. Élément purement illustratif.
Message		Élément décoratif utile pour aider le lecteur du diagramme à visualiser un flux de message.
Regroupement (Group)		Permet d'entourer des éléments afin de les regrouper visuellement à l'intérieur d'un processus. Cela n'a aucun impact du point de vue de l'exécution.
Annotation		Permet d'annoter les diagrammes de manière standardisée afin de pouvoir être exporté dans le fichier bpmn 2.0 au format XML.
Artefact personnel		La norme BPMN définit la possibilité d'ajouter des artefacts décoratifs personnels. Les modélisateurs les intègrent avec les mêmes propriétés et attributs que les autres artefacts.

Tableau 7: Éléments Artefacts

Les artefacts sont des éléments facilitant la lecture des diagrammes en permettant d'ajouter des informations additionnelles pertinentes pour certains corps de métier. Ces éléments n'ont aucun impact sur le déroulement du processus et sont inutiles du point de vue de l'exécution par un moteur de workflow. Les artefacts sont toujours associés à un élément de flux via un lien de type association (voir tableau 4). Ci-dessous un exemple d'utilisation des artefacts.

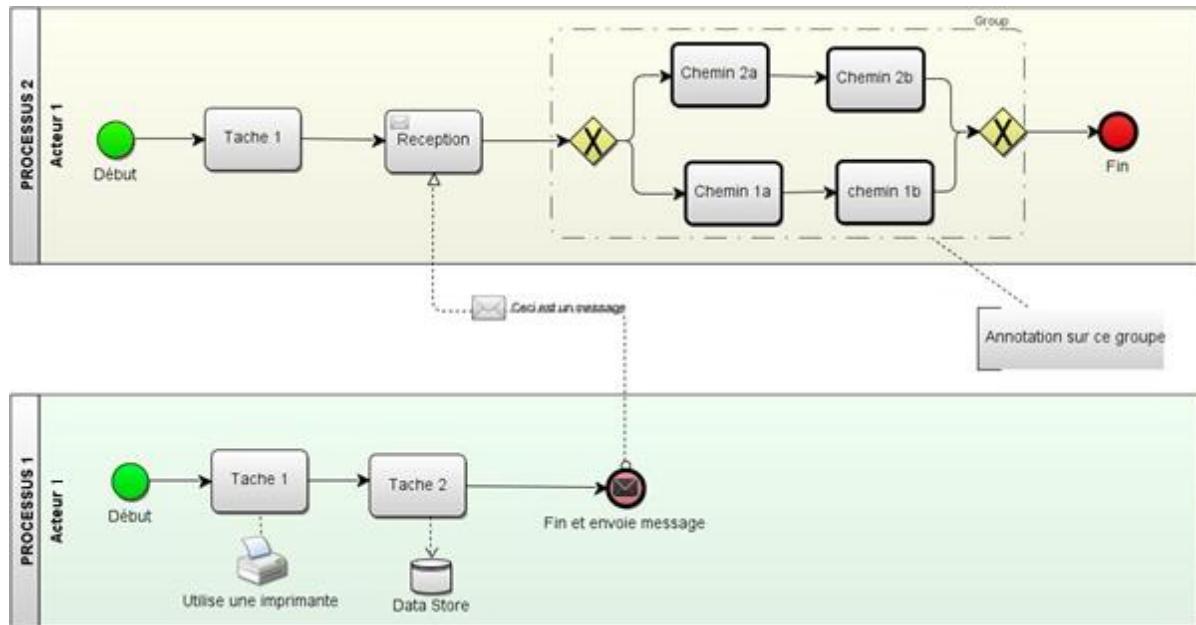


Figure 59: illustration, processus avec artefacts

Nous avons présenté les 4 catégories d'éléments permettant de modéliser des processus avec la norme BPMN 2.0. Bien qu'il existe de nombreuses catégories d'éléments BPMN, il faut comprendre que tous sont des symboles dérivés des représentations graphiques issues des 4 catégories précédentes. Il suffit de connaître les principes de base et

quelques dérivations usuelles pour pouvoir commencer à modéliser avec BPMN. En effet, la plupart des éléments décrits dans la norme sont très spécifiques et ne servent que certains cas particuliers. Rares sont les logiciels de modélisation fournissant toute la palette des éléments BPMN 2.0. En pratique une trentaine d'éléments sont très courants et suffisent à représenter un modèle même exécutable. Au point suivant nous présenterons les différents types de diagrammes définis par la norme et aborderons la décomposition hiérarchique de ces derniers.

A4-3. Principes de modélisation

A4-3.1 Les différents type de diagrammes

À l'aide d'éléments issus des 4 catégories précédentes il est possible de modéliser des diagrammes de description de processus. Il existe 4 sortes de diagrammes différents définis par la spécification de l'OMG.

- Les diagrammes de processus Privé ou Public
- Les diagrammes de Collaboration
- Les diagrammes de Chorégraphie
- Les diagrammes de Conversation

Les diagrammes de processus Privé ou Public.

Ce sont les représentations classiques d'un processus qui permettent de représenter un processus spécifique à une organisation en précisant les sous-processus, les activités ou les tâches, les passerelles, les événements, les objets échangés. Nous les nommons *processus privés* lorsqu'ils contiennent un seul bassin²⁷ (exemple *figure 57*). Le flux séquentiel des activités ou des tâches du processus est contenu dans les limites du bassin et ne peut pas les traverser. Le *processus public* quant à lui est un processus privé plus des interactions avec un ou plusieurs Participants en définissant les flux de messages, leur séquence, leur ordre etc. Seules les activités de communication avec l'autre(s) participant sont présentées dans le processus public. Toutes les autres activités ou tâches internes du processus cible ne sont pas représentées. Ci-dessous la *figure 60* est l'exemple présenté par l'OMG d'un processus public.

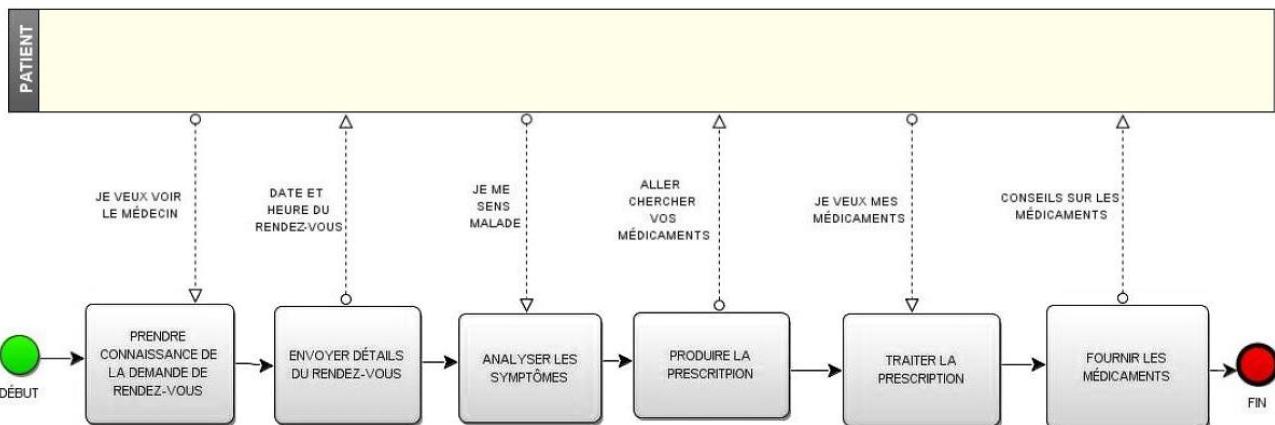


Figure 60: Exemple d'un processus Public (OMG, 2013, p.24) [traduit en français et coloré].

Les diagrammes de Collaboration

Le diagramme de collaboration permet de représenter deux processus privés ou plus et leurs interactions. Les Bassins sont définis comme étant les participants de cette collaboration. Les messages échangés entre les participants du diagramme de Collaboration sont présentés à l'aide du symbole flux de message. Ce symbole permet de connecter les bassins entre eux (ou les objets que l'on retrouve à l'intérieur de ces bassins). Voir *figures 58* ou *figure 59* pour un exemple de Collaboration.

²⁷ Lorsque le processus ne contient qu'un type d'acteur (donc aucun couloir) il est aussi possible de ne pas placer notre processus dans un bassin.

Les diagrammes de Chorégraphie

Une Chorégraphie est la modélisation d'un comportement attendu entre des Participants qui interagissent les uns avec les autres et qui veulent coordonner leurs activités ou leurs tâches à l'aide de Messages. Dans ce type de modélisation, la focalisation n'est pas sur l'Orchestration (processus public ou privé), c'est-à-dire sur la manière dont est accompli le travail selon le point de vue des participants, mais sur les échanges de Messages entre les Participants. Alors qu'un processus public ou privé est représenté à l'intérieur d'un bassin, une chorégraphie est représentée entre les bassins (ou les participants). Une Chorégraphie ressemble à un processus privé dans la mesure où elle se présente sous la forme d'un réseau d'Activités, d'Événements et de Passerelles. Cependant, une Chorégraphie en diffère puisque les Activités en interactions représentent des échanges de Messages impliquant deux ou plusieurs Participants. Une activité de chorégraphie n'est jamais positionnée dans un bassin ou un couloir, les participants sont représentés d'une manière propre à ce modèle (OMG, 2013, p.25). Les diagrammes de Chorégraphie utilisent un élément spécifique « les tâches de chorégraphie²⁸ » (choreography task) spécifiant un participants en haut et en bas de l'élément. Celui sur fond clair est l'émetteur du message, celui sur fond sombre le récepteur.

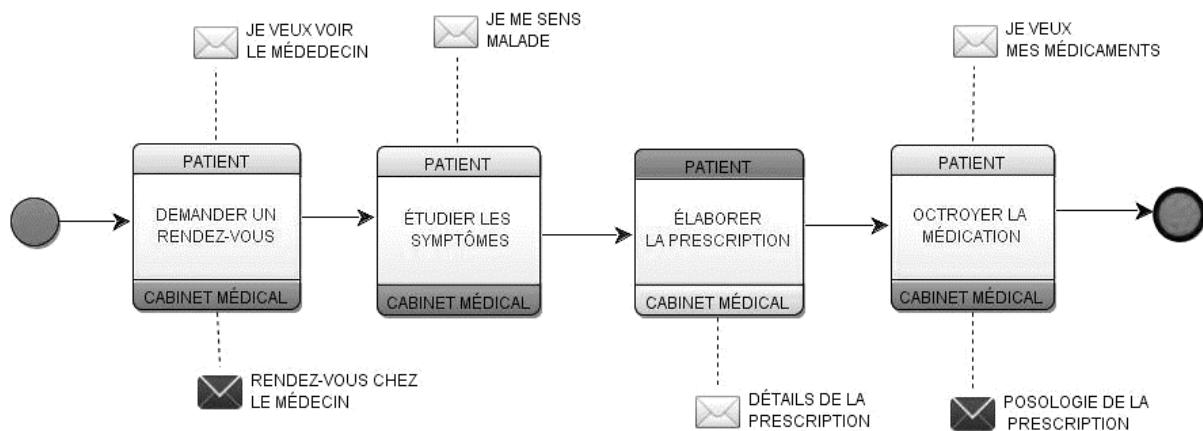


Figure 61: Exemple de diagramme de Chorégraphie (OMG, 2013, p.25) [traduit]

Les diagrammes de Conversation

Le diagramme de Conversation est la description informelle d'un diagramme de collaboration à haut niveau. Il représente des échanges de Messages (présentés sous la forme d'un regroupement de flux de messages), logiquement reliés entre les participants et qui concernent un objet métier d'intérêt, par exemple dans le domaine des affaires, un ordre, un envoi, une livraison ou une facture. Il représente un ensemble de flux de messages qui sont regroupés ensemble. Une Conversation peut impliquer deux ou plusieurs participants.

- Un diagramme de Conversation représente les conversations comme des hexagones, entre les participants (les bassins). Si l'on clique sur l'hexagone, on peut avoir le détail des flux de messages échangés entre les Participants.
- Les bassins qui servent à la représentation d'un diagramme de Conversation, ne contiennent habituellement pas de processus et sont donc présentés vides.
- Aucun diagramme de Chorégraphie ne peut être placé entre les bassins du diagramme de conversation.

²⁸ Notons que comme tous les autres éléments « tâche » de BPMN, l'élément a pour symbole de base un rectangle aux coins arrondis.

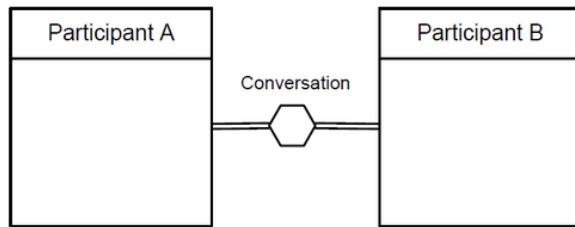


Figure 62: Représentation simple d'un diagramme de Conversation BPMN

A4-3.2 Décomposition hiérarchique, boucles et multi-instanciation

Décomposer une modélisation en sous-processus

Un processus important peut rapidement devenir complexe. Il est alors possible de décomposer le processus en sous-processus afin de pouvoir garder une vision d'ensemble (macro-processus). De plus ce type de modélisation permet

- de réutiliser à volonté des sous-processus dans différents macro-processus sans avoir à les ré-modéliser.
- D'orchestrer le déroulement des sous-processus (déclenchement d'un processus selon un événement, lancement de processus en parallèles)

Un sous-processus est un élément tâche particulier reconnaissable par un signe « + » à sa base. Lorsque l'on clique sur cet élément il alors possible de voir le processus qu'il contient (figure 63).

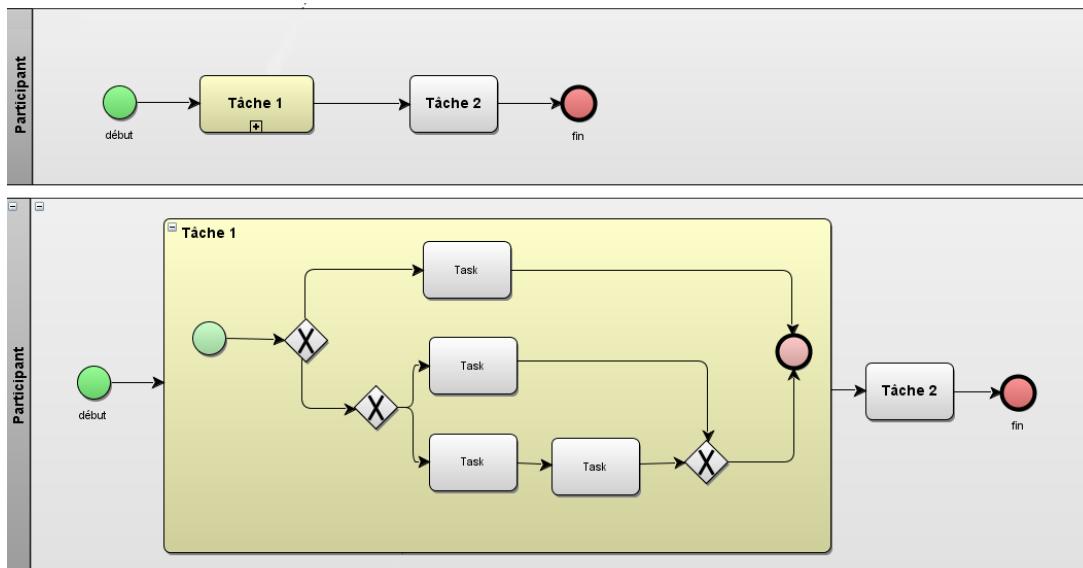


Figure 63: Ouverture d'un sous-processus lors d'un clic sur l'élément.

Les éléments présents dans un sous-processus n'interagissent pas directement avec les éléments du processus supérieur. Un sous-processus se réalise indépendamment. Il s'agit d'un processus distinct avec sa propre séquence d'exécution dont le résultat d'exécution permet d'effectuer les activités subséquentes du processus supérieur. La norme définit un sous-processus particulier nommé « Sous-processus Ad-Hoc » permettant de définir un sous processus dont les tâches internes ne sont pas organisées selon un workflow défini. Cet élément est reconnaissable via le « ~ » à sa base.



Les tâches sont alors effectuées de manière séquentielle ou parallèle. Ci-dessous, deux sous-processus Ad-Hoc permettant de définir les tâches nécessaires à l'écriture d'un chapitre sans en spécifier le « flux ». À gauche les tâches seuls, à droite les mêmes tâches avec l'ajout d'artefacts spécifiant les intrants et extrants des tâches.

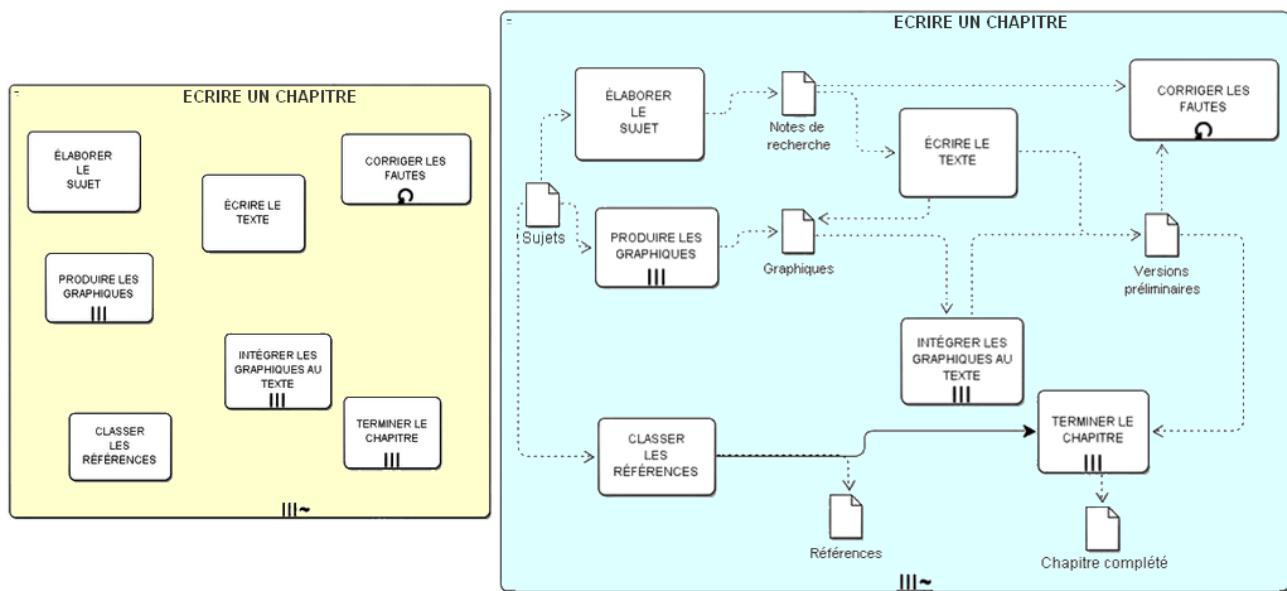
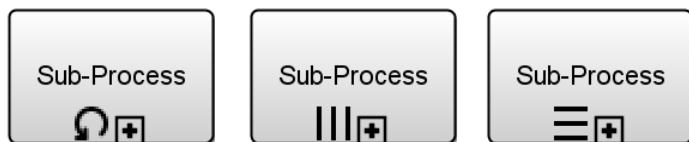


Figure 64: Processus Ad-Hoc (OMG, 2013, p.182-183) [traduit et coloré]

Boucles et multi-instanciation

Sur les tâches de la figure 64 nous pouvons voir apparaître des sigles en bas des rectangles symbolisant les tâches. Ces sigles représentent graphiquement la valeur de l'attribut « loopCharacteristics » associé à chaque élément de type « tâche » (tel que vu précédemment tableau 2). Une flèche pivotant sur elle-même indique une activité de réitération, la tâche se réalise « tant que » la condition définie se réalise. Les trois traits indiquent quant à eux une tâche d'instance multiple correspondant à la forme algorithmique « pour chaque ». Lorsque les lignes sont horizontales, les tâches sont effectuées en parallèle du processus et lorsqu'elles sont verticales les activités sont menées de façon séquentielle.

Les sous-processus étant des éléments de type « tâche », ils possèdent l'attribut graphique de boucle et instance multiple, comme tous les autres éléments de ce type.



ANNEXE 5 – EXEMPLE WSDL 2.0

```

<?xml version="1.0" encoding="UTF-8"?>
<description xmlns="http://www.w3.org/ns/wsdl"
    xmlns:tns="http://www.tmsws.com/wsdl20sample"
    xmlns:whttp="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/"
    targetNamespace="http://www.tmsws.com/wsdl20sample">

    <!-- Abstract type -->
    <types>
        <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns="http://www.tmsws.com/wsdl20sample"
            targetNamespace="http://www.example.com/wsdl20sample">

            <xs:element name="request"> ... </xs:element>
            <xs:element name="response"> ... </xs:element>
        </xs:schema>
    </types>

    <!-- Abstract interfaces -->
    <interface name="Interface1">
        <fault name="Error1" element="tns:response"/>
        <operation name="Opp1" pattern="http://www.w3.org/ns/wsdl/in-out">
            <input messageLabel="In" element="tns:request"/>
            <output messageLabel="Out" element="tns:response"/>
        </operation>
    </interface>

    <!-- Concrete Binding Over HTTP -->
    <binding name="HttpBinding" interface="tns:Interface1"
        type="http://www.w3.org/ns/wsdl/http">
        <operation ref="tns:Get" whttp:method="GET"/>
    </binding>

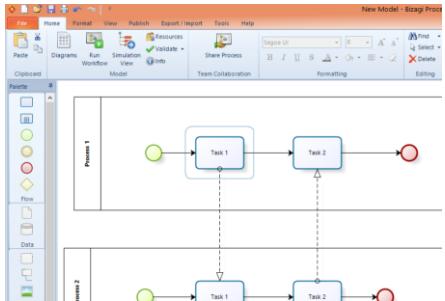
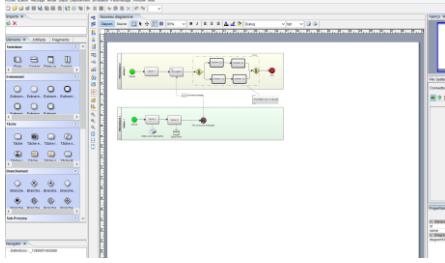
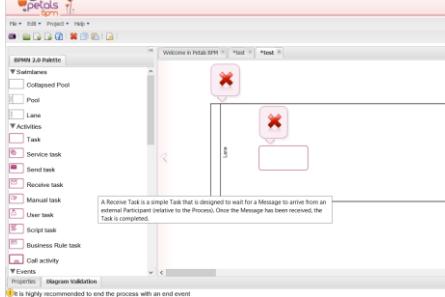
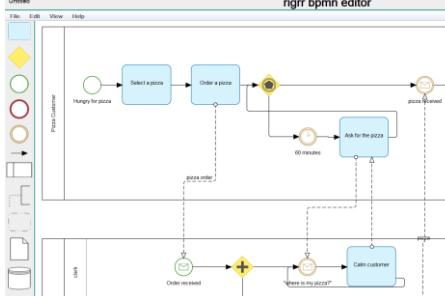
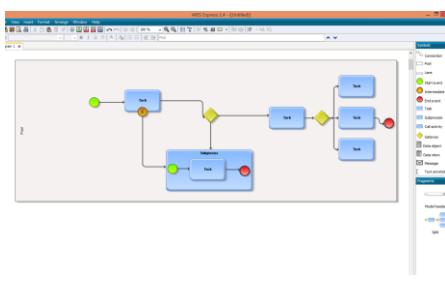
    <!-- Concrete Binding with SOAP-->
    <binding name="SoapBinding" interface="tns:Interface1"
        type="http://www.w3.org/ns/wsdl/soap"
        wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/"
        wsoap:mepDefault="http://www.w3.org/2003/05/soap/mep/request-response">
        <operation ref="tns:Get" />
    </binding>

    <!-- Web Service offering endpoints for both bindings-->
    <service name="Service1" interface="tns:Interface1">
        <endpoint name="HttpEndpoint"
            binding="tns:HttpBinding"
            address="http://www.example.com/rest/"/>
        <endpoint name="SoapEndpoint"
            binding="tns:SoapBinding"
            address="http://www.example.com/soap/"/>
    </service>
</description>

```

Figure 65: Exemple illustratif d'un document WSDL 2.0 (extrait Wikipédia)

ANNEXE 6 – QUELQUES OUTILS DE MODÉLISATION GRATUITS

Aperçu	Description/ Avis
	BIZAGI Process Modeler
	Yaoqiang Process editor 3.0
	Petals BPM
	Rigrr BPMN editor
	Aris Express

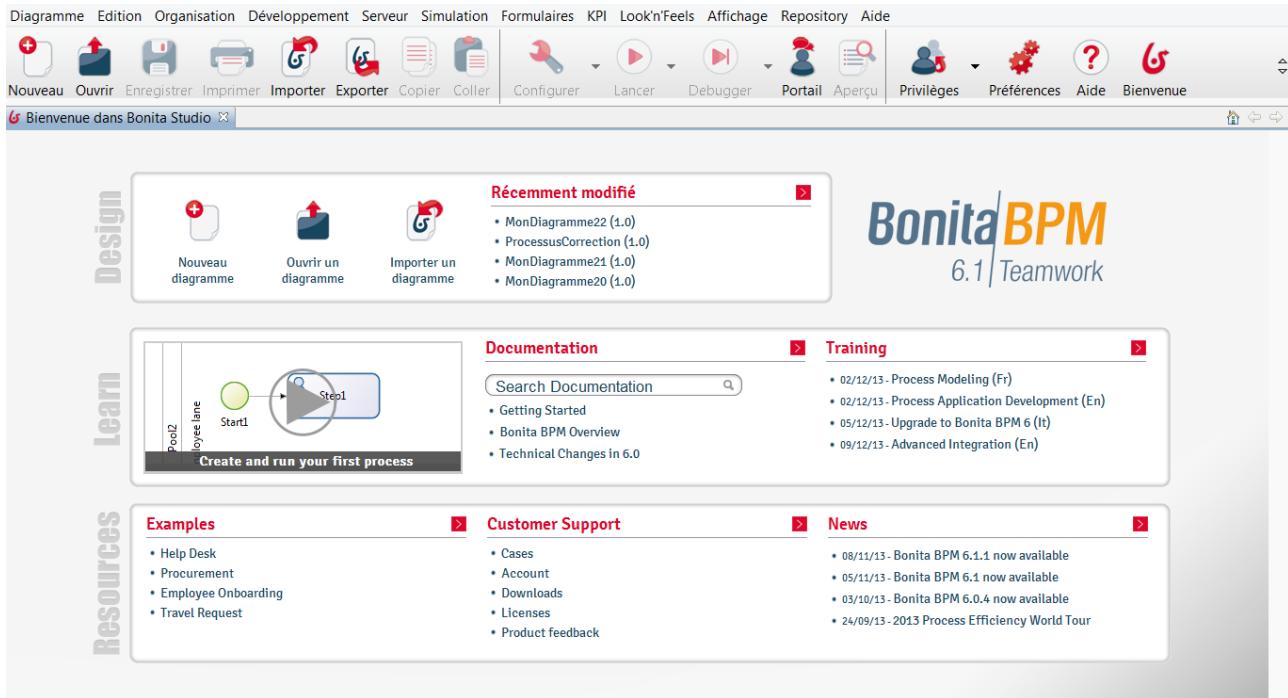
ANNEXE 7 – BPMS PROPOSANT UNE SOLUTION GRATUITE

« Approche » signale « une volonté de ». Dans les faits tous les BPMS demandent des phases de développement de code. Voir chapitre 4.

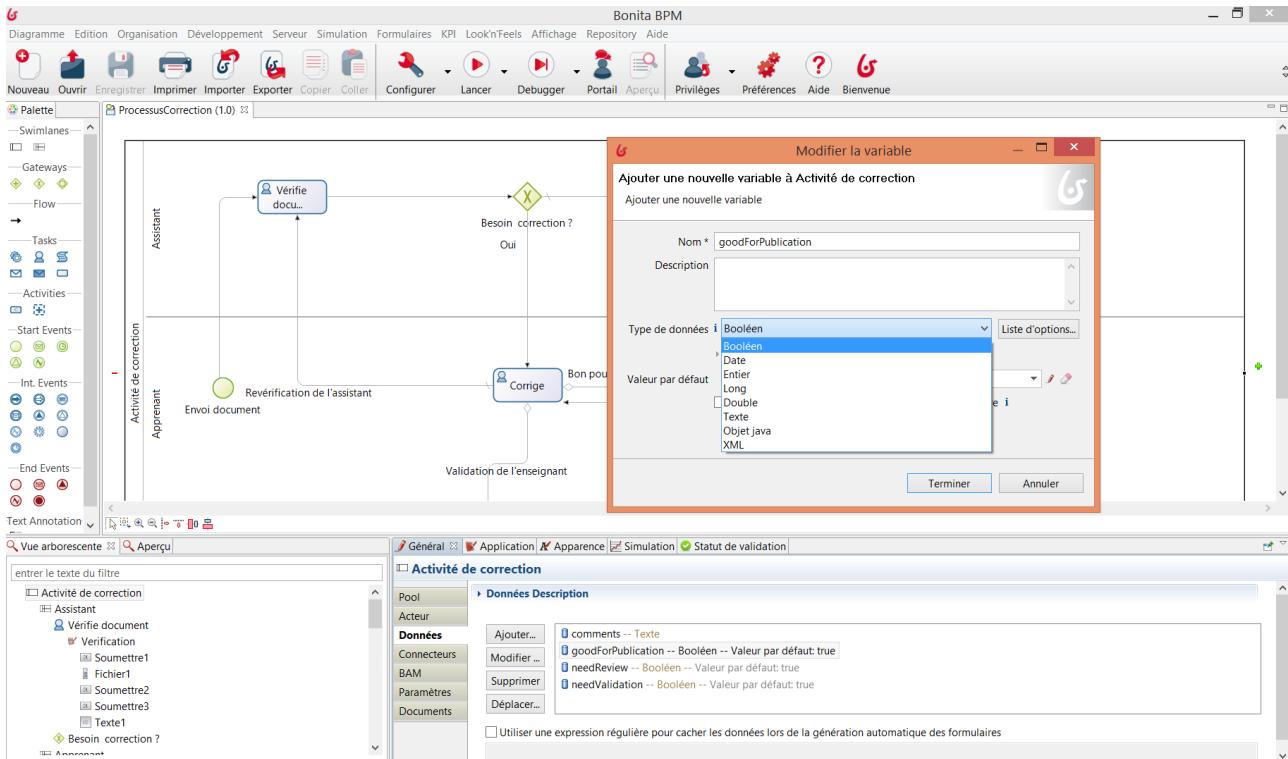
Nom / Développeur	Licence	Approche	Description / Fonctionnalités	Avis
Bonitasoft BPM	Version "Community" gratuite GNU General Public License v2.	Zero code	Bonita BPM intègre un modélisateur, un moteur de workflow Java performant et des connecteurs relativement simple à mettre en œuvre pour interfaçer les processus avec des services. Bonita permet ensuite le déploiement via une plateforme d'administration web autonome et conçu pour s'intégrer avec les environnements existant déjà dans les organisations.	Certainement l'une des meilleures suites libres pour la modélisation et l'exécution de BPMN. Bonita se développe de manière dynamique avec une des approches « zero-coding » les plus avancées. Cependant la version « community » n'offre pas toutes les fonctionnalités, notamment concernant la gestion des KPI, la modification du design de la plateforme où la synchronisation LDAP.
Activiti	Open source Licence Apache	Code	Activiti BPMS est une suite d'applications très légère conçue pour s'intégrer aux outils déjà utilisés par les développeurs et business man en entreprise. Elle comprend un modélisateur dans une interface web basée sur Signavio, Un plugin de développement basé sur Eclipse, un programme d'exécution, un explorateur web pour déployer le processus et divers outils en ligne pour collaborer et intégrer les documents déjà présent en entreprise dans le processus.	Activiti BPM Platform se présente comme une suite complète allant de la modélisation à l'implémentation et aux suivis de processus métier. Cette suite n'est pas appropriée aux personnes non techniques. Activiti est destiné à créer des processus "Rock solid" en facilitant la communication entre business man et développeurs Java, et en s'intégrant aux programmes déjà présents en entreprise. Activiti à l'avantage d'être léger et puissant dans les mains d'expert informatique.
Bizagi BPM Suite	Logiciel propriétaire. « Xpress edition » gratuit. « Enterprise edition » payante.	Less code	Modélisation, connexion aux outils (messagerie, base de données), exécution de l'application web autonome, choix du design de l'interface finale, portail permettant le suivi et la génération de rapport sur le processus.	Application professionnelle puissante avec une très bonne interface facile à utiliser. Bonne documentation. Un produit qui semble intéressant mais nous ne l'avons pas assez testé pour savoir s'il est facile d'exécuter et implémenter un processus.
Intalio BPMS	Community edition : gratuit et Enterprise Edition : Payant et 100% open-code	Zero code	Une suite logicielle composée de trois parties (Intalio Designer, Intalio Server et Intalio Workflow). Elle propose une génération automatique du code, sans programmation et une grande compatibilité avec les serveurs d'applications (WebLogic, WebSphere, Geronimo, JBoss) et les bases de données (Oracle, MySQL, MS SQL Server...).	Bpmns designer est une suite logicielle très graphique développée autour de l'idée "One-Click Process Deployment" poussant au maximum la volonté d'un outil "zero-coding". Dans les faits, ce n'est pas toujours le cas son utilisation se révèle assez semblable à Bonita BPM. L'inscription sur le site officiel est nécessaire pour avoir accès à une bonne documentation.
Sydle Seed	Freeware version payante (options d'intégrations supplémentaires, utilisateurs illimités, personnalisation de l'apparence)	Code	Modélisation, exécution et gestion de processus simples via une solution totalement en ligne. Suivi au moyen d'indicateurs de processus, Tableau de bord personnalisable.	Suite Bpmn nouvelle sur le marché codée en Java et totalement accessible en ligne. Créée par une entreprise brésilienne qui semble assez méconnue en Europe. Un produit qui semble intéressant mais la documentation est quasi inexistante sur le web. De plus la manipulation avancée de données passe par un langage de script Java (MVEL) et la connexion à des services externes est limitée.
JBoss community JBPM	free open-source	code	jBPM5 permet d'exécuter des processus définis en utilisant le format BPMN 2.0. JBPM5 propose différents outils pour modéliser, exécuter, gérer et contrôler les processus métier. Deux éditeurs sont proposés l'un s'intégrant dans l'IDE Eclipse sous la forme d'un plugin, l'autre est un outil cloud basé sur le navigateur.	Approche très similaire à Activiti (Les fondateurs d'Activiti viennent de JBoss). JBPM est un outil puissant à réservoir aux utilisateurs sachant coder en Java. Léger, totalement open-source avec une très bonne documentation.

ANNEXE 8 – CAPTURE D’ÉCRAN DE BONITA BPM 6

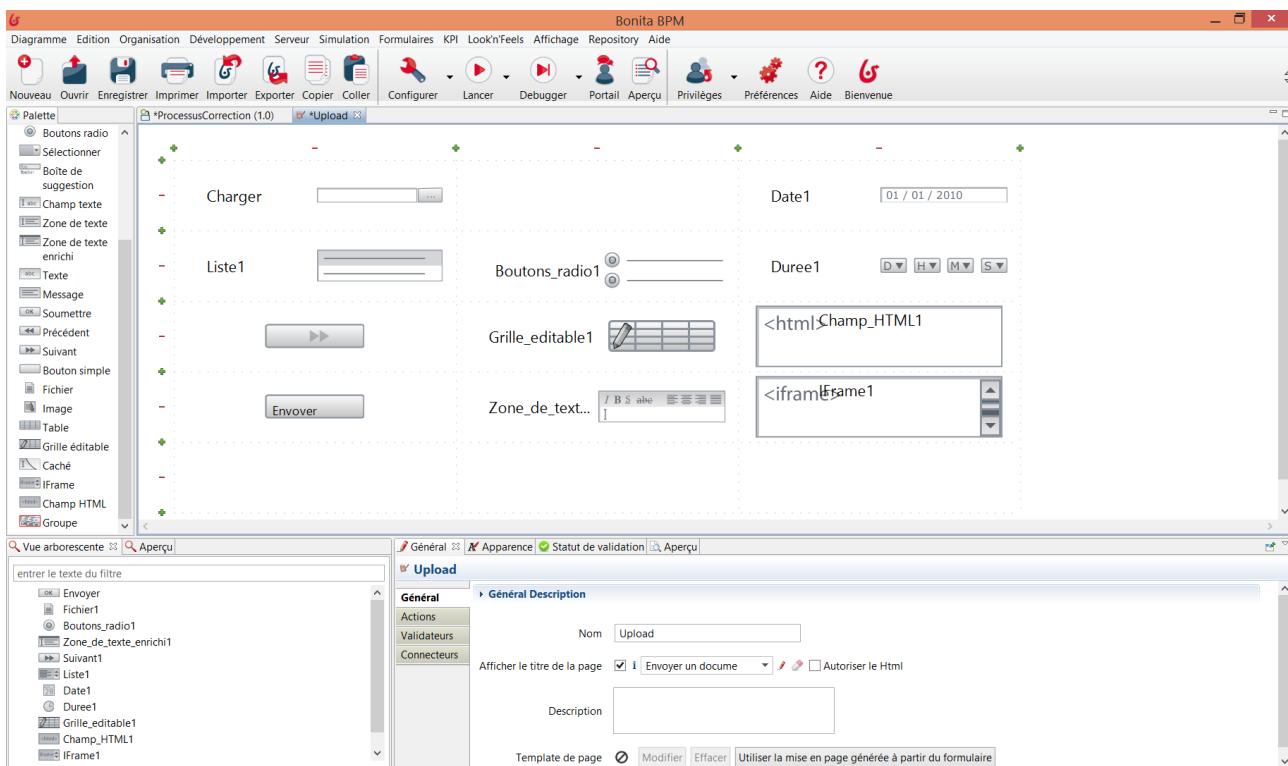
A7-1 – Accueil (type écosystème)



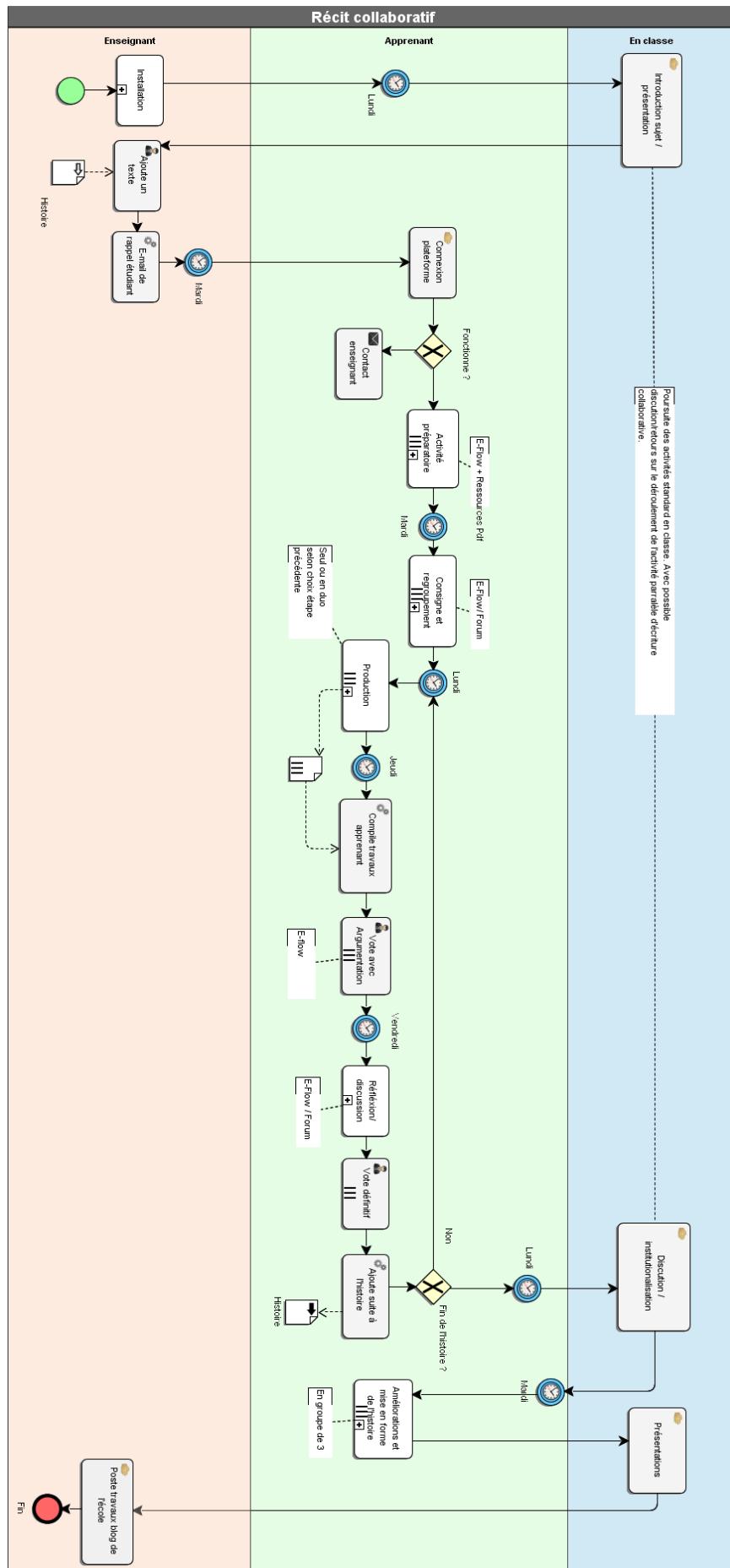
A7-2 – Aperçu de l’interface de développement (définition de variable)



A7-3 Aperçu création d'interface formulaireire



ANNEXE 9 – SCÉNARIO BPMN RÉCIT COLLABORATIF



ANNEXE 10 – DÉTAILS PROCÉDURE DE DÉVELOPPEMENT

Un scénario doit être modélisé avec BPMN soit dans le Studio Bonita soit avec un modélisateur externe conforme BPMN 2.0. Dans ce cas le diagramme devra être importé via le format standard BPMN 2.0 dans le Studio Bonita. Cette opération s'est plus ou moins bien passée selon nos essais. Pour qu'une importation réussisse il faut :

- Utiliser un modélisateur de qualité. C'est-à-dire respectant le standard et évitant l'ajout de fonctionnalités personnelles.
- Faire valider son diagramme par l'outil.
- Éviter les modélisations avancées récentes de BPMN 2.0. Si elles sont définies dans la norme, cette dernière est encore très jeune et les outils l'implémentent encore de manière inégale.
- Garder à l'esprit que Bonita est un outil pour modéliser des diagrammes d'implémentation et non de communication (La palette d'élément de Bonita ne contient pas l'entièreté des éléments définis par la spécification).

Une fois le diagramme défini dans Bonita studio ou importé depuis un outil externe. Nous avons suivi la procédure suivante pour l'implémentation et l'installation sur E-flow.

Développement :

1. Création de variables pour la gestion des données et des mécanismes conditionnelles. Les variables peuvent être créées au niveau du processus global (Bassin) ainsi l'ensemble des éléments qu'il contient y ont accès. Elles peuvent aussi être définies au niveau local (tâches) dans ce cas seule la tâche et ses connexions sortantes y ont accès.
2. Complétion des attributs nécessaires à l'exécution pour chaque élément graphique du diagramme. En particulier les conditions pour chaque flèche BPMN.
3. Définition des acteurs pour chaque couloir du diagramme BPMN. Des filtres d'acteur préinstallés ou des scripts doivent être utilisés pour la sélection d'acteur avancé.
4. Création des écrans d'interface pour chaque tâche utilisateur. La création s'effectue via une interface drag & drop permettant de créer des formulaires pour chacune des étapes afin que les utilisateurs finaux puissent uploader les documents, ajouter des commentaires, faire des choix parmi une liste d'options proposées etc. Les différents champs du formulaire doivent ensuite être liés aux variables définis précédemment.
5. Utilisation d'un design Bonita ou création d'un nouveau via feuilles de style CSS ou l'interface graphique (licence payante) pour personnaliser les formulaires.
6. Utilisation et paramétrage de connecteurs pré-intégrés pour la connexion à des services externes (ex : communication avec Gmail).
7. Exportation de l'application en un fichier .bar exécutable sur E-flow.

Il faut parfois ajouter des éléments de configuration serveur et l'écriture ponctuelle de quelques lignes de codes pour effectuer des requêtes ou des opérations spécifiques. Enfin il est possible de définir des KPI pour l'analyse de données (voir section 1.3 de ce chapitre). Ces derniers seront attachés au processus et permettront de collecter des données à chaque instance du processus pour la génération de rapport.

Installation sur E-flow :

1. Charger le fichier .bar depuis l'onglet « Gestion des application » du mode administrateur.
2. Choisir des groupes d'utilisateur pour l'opération de « mapping » afin que le programme attribut les bons utilisateurs à chaque couloirs BPMN.
3. Remplir les paramètres de départ (Si il y en a).
4. Activer l'activité. Tant que l'activité n'est pas activée les utilisateurs n'ont pas accès à l'activité.

ANNEXE 11 – SOFT-SKILLS COURSE - JIŘÍ KOLÁŘ, TOMÁŠ PITNER

