



**INSTITUT DE LA FRANCOPHONIE
POUR L'INFORMATIQUE**

RAPPORT DU TIPE

(Rapport final)

PRÉSENTÉ PAR

TIEU KIM CUONG

LES NORMES POUR LA FORMATION À DISTANCE

SOUS LA DIRECTION DE

Dr. NGUYEN HONG QUANG

Hanoi, novembre 2006

REMERCIEMENTS

Cette étude a été réalisée avec l'aide de plusieurs personnes. D'abord je remercie Dr. NGUYEN Hong Quang, professeur à l'IFI, qui me donne des idées utiles et m'oriente dans toutes les étapes des études contribuant essentiellement à ce succès.

Je remercie également mes amies qui m'aident dans ce travail avec des documents, des idées, des conseils, etc.

Enfin, je remercie aussi tous les professeurs qui me donnent des cours jusqu'au moment.

Hanoi, le 2 novembre 2006

TIEU Kim Cuong

Promotion 11 à l'IFI

LES ABRÉVIATIONS

N°	Mots abrégatifs	Groupe de mots en réalité
1	ADL	Advanced Distributed Learning
2	AICC	Aviation Industry CBT Committee
3	API	Application Program Interface
4	ARIADNE	Alliance of Remote Instructional Authoring & Distribution Networks for Europe
5	CBI	Computer-Based Instruction
6	CBT	Computer-Based Training
7	CMI	Computer Managed Instruction
8	CSF	Content Structure Format
9	DoD	Department of Defense
10	DTD	Document Type Definition
11	FAD	La Formation à Distance
12	FOAD	La Formation Ouverte et à Distance
13	HTTP	Hypertext Transfer Protocol
14	IDA	Institute for Defense Analyses
15	IEEE	Institute of Electrical and Electronics Engineers
16	ISO	International Organization for Standardization
17	ITS	Intelligent Tutoring Systems
18	LCMS	Learning Content Management System
19	LMS	Learning Management System
20	LO	Learning Object
21	LOM	Learning Objects Metadata
22	LTSC	Learning Technology Standards Committee
23	SCO	Sharable Content Object
24	SCORM	Sharable Content Object Reference Model
25	URI	Universal Resource Identifier
26	URL	Universal Resource Locator
27	XML	eXtensible Markup Language

TABLE DES MATIÈRES

<i>Introduction</i>	8
<i>Chapitre 1 Introduction à la formation ouverte et à distance (FOAD)</i>	11
1.1. Définitions	11
1.2. Histoire de la formation ouverte et à distance	11
1.3. FOAD dans l'ère de l'Internet.....	13
1.4. Plates-formes FOAD (<i>Campus numériques</i>).....	13
1.5. Problème de partage de ressources	15
1.6. Standards et normes pour la FOAD	15
1.6.1. Notion de norme et standard	15
1.6.2. Modèle SCORM et ses avantages.....	16
<i>Chapitre 2 Etude de cas : le modèle SCORM</i>	17
2.1. Histoire brève du modèle SCORM.....	17
2.2. Modèle SCORM 2004	19
2.2.1. Introduction au modèle SCORM 2004	19
2.2.2. Modèle d'agrégation du contenu (CAM)	20
2.2.3. Environnement d'exécution (RTE)	24
1) Modèle de données de SCORM RTE.....	25
2) Fonctions de l'API.....	25
2.2.4. Séquencement et la navigation du SCORM (SN)	26
1) Quelques notions correspondantes	27
2) Modes de contrôle du séquencement (Sequencing Control Modes- SCM)	28
3) Règles de séquencement (Sequencing Rules - SR).....	28
<i>Chapitre 3 Processus de transformation d'un cours en ligne à la norme de SCORM 2004</i>	30
3.1. Etape préparatoire	30
3.2. Etape de transformation d'un cours local à la norme de SCORM 2004	31
3.2.1. Etude de la structure d'un cours en norme de SCORM 2004	31
3.2.2. Etudes des étapes de transformation d'un cours local à la norme de SCORM 2004	32
1. Etude de la transformation des fichiers du cours aux SCO.....	32
2. Etude de la construction d'un Paquet de SCORM 2004.....	37
3.3. Etape d'importation sur une plate-forme choisie	39
3.3.1. Choix d'une plate-forme.....	40
3.3.2. Importation d'un paquet de contenus en norme de SCORM 2004 sur la plate-forme Moodle.....	43
<i>Chapitre 4 Transformation automatique d'un cours local à la norme de SCORM 2004 en utilisant l'éditeur Reload</i>	44
4.1. Étude d'éditeur Reload.....	44

4.2. Etude technique du programme à modifier	46
4.2.1. Description	46
4.2.2. Scénario d'utilisation du programme	46
4.2.3. Insertion automatique des fonctions aux SCO	47
4.2.4. Fenêtre principale du programme modifié	48
4.3. Processus de la transformation automatique d'un cours local à la norme de SCORM 2004 en utilisant l'éditeur Reload modifié	49
4.3.1. Transformation des fichiers du cours aux SCO	49
4.3.2. Construction d'un Paquet de SCORM 2004	50
4.3.3. Importation d'un Paquet de SCORM 2004 sur Moodle	51
<i>Conclusion</i>	52
<i>BIBLIOGRAPHIE</i>	54
<i>ANNEXE A Les deux librairies du programme</i>	57
<i>ANNEXE B Les éléments du modèle de données de SCORM RTE.....</i>	71

TABLE DES FIGURES

Figure 1.1 : Architecture de la e-Formation.....	14
Figure 2.1 : Processus développement du modèle SCORM	17
Figure 2.2 : Chronologie du SCORM	19
Figure 2.3 : L'organisation du modèle SCORM 2004	19
Figure 2.4 : Des Assets (a) et des SCO (b et c)	21
Figure 2.5 : Organisation de contenu	22
Figure 2.6 : Conditionnement du contenu (SCORM Content Packages)	23
Figure 2.7 : L'environnement d'exécution du modèle SCORM (RTE)	24
Figure 2.8 : Utilisation du Modèle de données avec l'objet d'API	25
Figure 2.9 : Relation entre un CO et un arborescence d'activité	26
Figure 2.10 : Arborescence et groupes conceptuels liés aux activités	27
Figure 3.1 : Modèle d'utilisation des SCO	31
Figure 3.2 : L'architecture en deux couches de SCO	33
Figure 3.3 : API fonctions, Objet d'API et Implémentation d'API	34
Figure 3.4 : Des transactions de communication entre LMS et SCO	35
Figure 3.5 : La structure du fichier imsmanifest.xml pour associer un Asset	38
Figure 3.6 : La structure du fichier imsmanifest.xml pour associer un SCO	39
Figure 4.1 : Fenêtre principale du Reload Editeur.....	44
Figure 4.2 : Les fonctions du Reload Editeur	45
Figure 4.3 : Les sous fonctions de la fonction Créer l'ADL SCORM 2004	45
Figure 4.4 : Scénario d'utilisation du programme.....	46
Figure 4.5 : Les deux fonctions ajoutées dans l'Editeur Reload	48
Figure 4.6 : Construction d'un SCO à partir des fichiers du cours	49
Figure 4.7 : La fenêtre de construction d'un paquet de SCORM 2004	50
Figure 4.8 : Importation des SCO dans le cours.....	51

RESUME

Dans le cadre de ces TRAVAUX D'INTÉRÊT PERSONNEL ENCADRÉ (TIPE) à l'IFI avec le sujet « LES NORMES POUR LA FORMATION À DISTANCE », nous allons présenter des études sur la formation ouverte et à distance (FOAD) et son problème de partage de ressources. Il y a plusieurs normes et standards établis, mais ils ne résolvent pas ce problème complètement. La naissance de la norme de SCORM (Sharable Content Object Reference Model) créé par ADL (Advanced Distributed Learning), qui est un ensemble de spécifications techniques corrélant les travaux des autres organismes de normalisation comme AICC, IMS et IEEE/LTSC, permet de réutiliser des contenus d'apprentissage de formation via le Web, à travers divers environnements et des plates-formes différentes supportant le format SCORM. Après avoir fait des études théoriques en détail sur ce modèle, nous proposons des processus pour transformer un cours en ligne à la norme de SCORM 2004 automatiquement en utilisant l'Editeur Reload modifié.

Introduction

Dans notre siècle, le développement rapide de Technologies de l'Information et de la Communication, notamment de l'Internet, rend la formation plus flexible et adaptative. La Formation à distance (FAD) joue un rôle de plus en plus important dans les domaines de l'apprentissage et de l'éducation.

Pour réaliser ce type de formation il nous faut mettre sur la table quelques questions importantes à résoudre :

1. Quels moyens va on utiliser pour cette formation ?

Pour résoudre ce problème il faut savoir l'histoire de la formation ouverte et à distance et son état actuel.

2. Comment peut on gérer toutes les activités d'apprentissage avec ces moyens choisis ?

C'est une grande question qui se pose des sous questions suivantes :

- 2.1 Comment peut on gérer des activités des professeurs, des apprenants et les interactions entre eux ?
- 2.2 Comment peut on représenter les contenus d'apprentissage ?

Pour résoudre ce problème il faut savoir l'architecture des « plates-formes » qu'on va utiliser, ainsi que des normes et standards pour présenter des ressources.

3. Comment peut on partager les ressources d'apprentissage entre les « plates-formes » conformant des normes et standards différents ?

Pour résoudre ce problème il faut trouver une norme qui corrèle avec des normes existantes. C'est la norme de SCORM.

4. Comment peut on présenter le contenu d'apprentissage à la norme de SCORM ?

Ce problème peut être divisé en deux sous questions :

- 4.1 Quelle version de SCORM peut on utiliser ?
- 4.2 Comment peut on présenter le contenu d'apprentissage sous la version choisie ?

Pour résoudre ce problème il y a deux choses à faire :

- **Partie théorique** : Il faut étudier en détail le modèle SCORM.
- **Partie pratique** : Il faut choisir un travail à faire sur ce modèle.

Dans ce travail nous avons étudié à résoudre tous les problèmes posés ci-dessus et nous avons divisé ce travail en des étapes suivantes :

1. Chercher tous les documents concernant la FOAD, la plate-forme, la norme et standard de la FOAD et les classifier.
2. Faire des études sur l'histoire de la FOAD.
3. Faire des études sur les plates-formes de la FOAD
4. Faire des études sur les normes et standards de la FOAD
5. Faire des études en détail sur le modèle SCORM
6. Réaliser une application sur le modèle SCORM. Dans cette étape nous avons choisi l'implémentation d'une application de transformation d'un cours en ligne à la norme de SCORM 2004.

Ce rapport se compose des chapitres suivants :

Chapitre 1 : Dans ce chapitre on présente l'introduction à la formation ouverte et à distance. On peut savoir qu'est ce que c'est la FOAD ? Son histoire et son état dans l'ère de l'Internet ; Qu'est ce que c'est une plate-forme pour la FOAD ? Son architecture et ses exigences de représentation des ressources ainsi que le problème de partage de ressources sur des plates-formes conformant des normes et standards

différentes; Quels sont des avantages du modèle SCORM pour résoudre ce problème ?

Chapitre 2 : Dans ce chapitre on présente en détail sur ce modèle SCORM. On peut savoir des processus développement du modèle SCORM ; Quelle est la version la plus stable de ce modèle ? Comment fonctionne le modèle SCORM 2004 ?

Chapitre 3 : Dans ce chapitre on présente des études sur les étapes en général de transformation d'un cours en ligne à la norme de SCORM 2004 et des problèmes qui nous défient de résoudre.

Chapitre 4 : Dans ce chapitre on présente des études sur les étapes en détail de transformation automatique d'un cours local à la norme de SCORM 2004 en utilisant l'éditeur Reload modifié et des problèmes qui nous défient de résoudre.

En conclusion nous présentons le bilan des résultats de ce travail, les leçons acquises à travers cette expérience et les suggestions éventuelles.

Chapitre 1

Introduction à la formation ouverte et à distance (FOAD)

1.1. Définitions

Enseignement à distance : C'est un type d'enseignement dans lequel les enseignants et les apprenants sont séparés par la distance géographique et le temps physique grâce à des médias de communication électronique ou imprimés ([27], p. 43).

Formation ouverte : C'est une formation flexible qui permet aux apprenants des entrées et des sorties permanentes ([3], p.2).

Formation à distance : C'est un système de formation conçu pour permettre à des individus de se former sans se déplacer sur le lieu de formation et sans la présence physique d'un formateur ([3], p.1).

Formation ouverte et à distance : C'est un dispositif souple de formation organisé en fonction de besoins individuels ou collectifs. Elle comporte des apprentissages individualisés et l'accès à des ressources et compétences locales ou à distance. Elle n'est pas exécutée nécessairement sous le contrôle permanent d'un formateur ([1], p.69).

E-Formation : C'est un sous-ensemble de la FOAD, qui s'appuie sur les réseaux électroniques. Aujourd'hui le concept d'e-Learning est de plus en plus employé, attestant de l'évolution fondamentale de ce domaine de formation ([3], p.3).

Plate-forme de formation (LMS): C'est un logiciel qui fournit aux trois principaux utilisateurs – formateur, apprenant, administrateur - un dispositif qui a pour premières finalités la mise à disposition, la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et le télé tutorat ([3], p.4).

1.2. Histoire de la formation ouverte et à distance

Ce type de formation existe depuis longtemps permettant aux apprenants de faire leurs études personnelles basées sur la notion de

l'**Auto-formation** accompagnée grâce aux moyens techniques et aux médias (par ex. la poste, la radio, la télévision, téléphone...) ([18], p.3-19) :

En 1840, l'année même de l'apparition du timbre-poste, création du premier cours de sténographie par Isaac Pitman. Très rapidement, cela a donné des idées à d'autres, et l'on voit apparaître un peu partout dans le monde occidental des initiatives concernant des formations pour lesquelles il n'y avait pas de réponse institutionnelle, entre autres les écoles de langues en Allemagne en 1856.

En 1873, aux États-Unis, la première société d'encouragement de l'étude à domicile est créée à Boston.

A Paris, pratiquement à la même époque, en 1877, M. Pigier crée ses fameux cours à vocation de formation professionnelle.

En 1880, première université par correspondance, toujours à Boston.

En 1885, apparition des cours préceptoraux de Hattemer à Paris au besoins d'accompagnement scolaire des enfants de la haute bourgeoisie parisienne. Puis cette méthode a été transposée à l'écrit permettant aux élèves en dehors de la présence d'un précepteur. Cela a très vite connu un développement important, et ces cours existent toujours aujourd'hui.

En 1891, naissance de l'École chez soi par Léon Eyrolles à Arcueil-Cachan permettant aux ouvriers de développer l'éducation en même temps que la formation professionnelle chez eux.

En 1913, Utilisation du cinéma pour l'éducation par Edison.

En 1921, la radio s'est perfectionnée, puisqu'il était possible d'envoyer un message à quelques dizaines ou centaines de kilomètres. Les premières radios ont été utilisées pour créer des cours universitaires et les diffuser aux étudiants malades ou qui ne pouvaient se déplacer.

En 1939, apparition des premiers cours par téléphone et par télévision via des programmes éducatifs concernant des étudiants malades ou ne pouvant pas se rendre à l'université.

En 1944, apparition du Centre national d'enseignement par correspondance, puis le Centre national de télé-enseignement, et aujourd'hui le Centre national d'enseignement à distance.

En 1946, création de la National Technical University, aux États-Unis, un dispositif permettant de faire du transfert rapide de la recherche vers les industries et les entreprises avec les cours donnés dans les troisièmes cycles

par des chercheurs à la radio. Très vite, elles sont devenues des émissions de télévision.

En 1946, le Funk Kolleg en Allemagne est devenu le Tele Kolleg avec diffusion des cours de secondaire.

En 1969, la première Open University britannique, l'ancêtre des universités ouvertes, est apparue.

Enfin en 1995, apparition d'une première université ouverte de Catalogne en Europe permettant aux parents d'élèves, aux enseignants et aux élèves d'échanger des informations entre eux.

1.3. FOAD dans l'ère de l'Internet

Le grand problème de la FOAD est toujours l'interaction entre les enseignants et les apprenants ([19], p.1-5 ; [20], p.81-82, 208-210). Mais, dans notre siècle, le développement rapide de l'Internet et ses technologies est en train de changer la façon de la formation, surtout de la FOAD ([24], p.3-5 ; [20], p.12-18 ; [21], p.1-2 ; [23], p.15). On peut apprendre n'importe quoi, n'importe quand et n'importe où sans difficulté d'échanger des informations ou discuter avec n'importe qui sur des problèmes qu'on a (avec des professeurs, des amies, des collègues, des gens utilisant l'Internet...), et la FOAD porte maintenant un nouveau nom : c'est e-Learning (e-Formation) qui est parlé dans tous le temps ([20], p.221-246 ; [22], p.1-3).

1.4. Plates-formes FOAD (*Campus numériques*)

Une plate-forme pour la formation ouverte et à distance est un logiciel qui fournit aux trois principaux utilisateurs – formateur, apprenant, administrateur - un dispositif qui a pour premières finalités la mise à disposition, la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et le télé tutorat ([3]).

On peut distinguer aujourd'hui trois types de plates-formes comme ci-dessous ([20], p.66):

□ **Les plates-formes « simples »** : Mises à disposition de cours en ligne, moyennes de communication asynchrones minimales, inscription directe à un ou des modules/cours.

□ **Les classes virtuelles synchrones** : Outils synchrones stimulant la classe, mais à distance : Une personne mène la présentation, avec un partage d'application, et permet aux auditeurs d'intervenir.

□ **Les plates-formes complexes** : Outre les fonctionnalités simples, ces plates formes intègrent soit un aspect d'ingénierie pédagogique, soit un aspect de médiatisation (intégration d'outils pour passer sous un format numérique du texte, une architecture de cours ...), soit encore un aspect de « compétences » (gestion des formations selon des évaluations de compétences et des profils de poste). Certaines plates-formes prennent en compte le support Internet uniquement. D'autres sont outillées pour gérer à la fois du « présentiel » et du « distanciel ». On parle alors de « mixe - formation ».

Maintenant on a environ 235 plates-formes existant dans tous le monde entier (Moodle, Atutor, ILIAS, Sakai...) ([4], p. 9). Le choix d'un de ceux dépend du but de la formation et de base de technique disponible. En général on peut voir le rôle d'une plate-forme dans un système FOAD comme suivant :

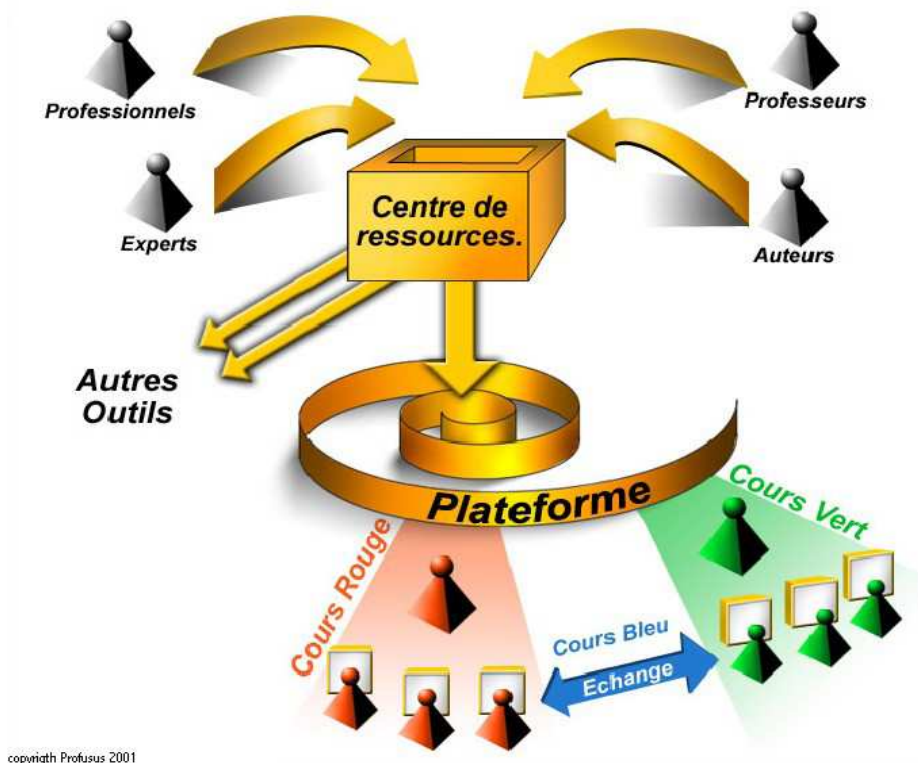


Figure 1.1 : Architecture de la e-Formation

1.5. Problème de partage de ressources

En réalité, on propose toujours des questions restant à résoudre comme suivantes ([4], p.10) :

- 1) Comment les contenus pédagogiques doivent-ils être décrits ?
- 2) Comment les informations des étudiants doivent-elles être enregistrées ?
- 3) Comment rendre transférable un cours d'une plate-forme à une autre?
- 4) Comment faire en sorte qu'une plate-forme intègre des ressources pédagogiques d'origines différentes ?

Ce sont des questions du problème de partage de ressources sur Internet entre les plates-formes différentes. Pour pouvoir résoudre ce problème il faut donner sur la table des normes et des standards pour présenter les contenus sur toutes les plates-formes existant et allant être créées dans l'avenir.

1.6. Standards et normes pour la FOAD

1.6.1. Notion de norme et standard

Les anglo-saxons n'utilisent que le terme standard pour les travaux de normalisations. Mais en France on distingue les deux termes un peu différentes : le standard et la norme. En général une norme est un document établi par consensus et approuvé par un organisme de normalisation reconnu (ISO, CEI UIT-T, ETSI ...) ([5]). Contrairement à une norme, un standard ne fait pas l'objet d'une publication qui en détaille le contenu ([5]). Un standard est un ensemble de recommandations développées et préconisées par un groupe représentatif d'utilisateurs et il est plus assimilé à un processus réactif de consensus du monde économique ou du monde technique ([6], p.10).

En réalité, il y a beaucoup d'Organismes internationales qui font des travaux de normalisation comme L'IEEE (*Institute of Electrical and Electronics Engineers*), l'IMS (*Instructional Management systems*), ARIADNE (*Alliance of Remote Instructional Authoring & Distribution Networks for Europe*), l'ADL (*Advanced Distributed Learning*), l'AICC (*Aviation Industry CBT committee*) ([7], p.20). Chaque Organisme développe ses normes et

ses spécifications basant ses buts. En fait, IMS développe le modèle de base de données, Méta-données pour les ressources d'apprentissage, Spécifications de contenu et d'assemblage, Conditionnement du contenu, Renseignements sur le séquençement ; AICC avec Structure du contenu ; IEEE/LTSC (*Learning Technology Standards Committee*) avec API IEEE 1484.11.2, Modèle de données de l'IEEE 1484.11.1, Spécifications LOM ([7], p.20-29) et **on a la difficulté de communiquer et de partager les objets d'apprentissage entre les plates-formes conformant des normes et standards différents.**

1.6.2. Modèle SCORM et ses avantages

Le modèle SCORM (*Sharable Content Object Reference Model*) est un ensemble de spécifications et de normes qui ont été rassemblées en une série de « documents techniques » ([9], p.15-16, 21, 37) et basées sur plusieurs normes et standards connus :

- les spécifications des méta-données de l'IEEE
- le modèle de données et l'API de l'IEEE
- la structure du contenu de l'AICC
- le conditionnement du contenu de l'IMS
- les renseignements sur le séquençement et le comportement de l'IMS ([9], p.37). Le modèle SCORM, qui combine les meilleurs aspects des normes d'AICC, d'IMS, d'LTSC et d'IEEE, donc, peut rendre les objets d'apprentissage **accessible** (approchable et disponible à utiliser), **adaptable** (adaptable aux besoins des apprenants), **durable** (indépendant de la technologie et héritable), **interopérable** (les contenus d'apprentissage peuvent être utilisés par n'importe quelle plate-forme ou outils FOAD) et **réutilisable** (une pièce du contenu d'apprentissage peut être utilisée dans plusieurs applications et contextes) ([9], p.32).

En plus, en faisant la comparaison entre des plates-formes connues dans le monde entier (61 plates-formes dans <http://www.edutools.net> en 2006) on trouve qu'il y a 42 plates-formes qui suivent des normes et standards (la plupart pour le modèle SCORM).

C'est la raison pour laquelle on utilise ce modèle dans ce travail.

Chapitre 2

Etude de cas : le modèle SCORM

2.1. Histoire brève du modèle SCORM

À l'origine de l'ordre exécutif 13111 chargé par DoD (*Le département de la défense aux Etats-Unis*) en 1997, le modèle SCORM (*Sharable Content Object Reference Model*), qui est un ensemble de spécifications techniques corrélant les travaux de AICC, IMS et IEEE/LTSC, a été créé par ADL (*Advanced Distributed Learning*) pour déterminer un modèle unique de contenu unifié pour la FOAD sur Internet. Ces spécifications permettent la réutilisation du contenu de formation via le Web, à travers divers environnements et produits, sous réserve que la plate-forme cible supporte le format SCORM. Le modèle SCORM permet la création de grains de cours, réutilisables et interopérables ([2], p.7 ; [8], p.13 et [9], p.15-16, 18, 27, 37-38).

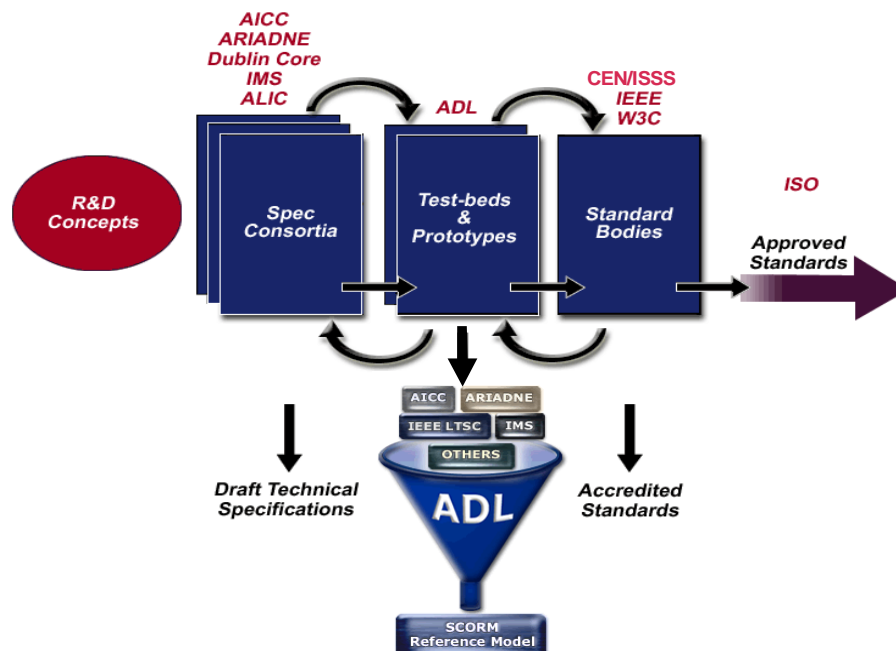


Figure 2.1 : Processus développement du modèle SCORM

Le modèle SCORM est une initiative continue et a évolué et a changé sensiblement depuis sa première version. Il y a actuellement trois versions principaux du SCORM ([11], p.8 et [10], p.5):

1) SCORM 1.1 (*Janvier 2001*)

Ceci a marqué la fin de la phase d'essai d'exécution et le début de la phase d'application pour l'ADL. Corrections incluses et améliorations de la version 1.1 de SCORM basées sur la rétroaction reçue du dégageant de la version 1.0 de SCORM. Le modèle en commun changeant de référence d'objet de support pédagogique (*Sharable Courseware Object Reference Model*) au modèle content en commun de référence d'objet (*Sharable Content Object Reference Model*) est le changement le plus apparent.

2) SCORM 1.2 (*Octobre 2001*)

La version 1.2 de SCORM a donné la capacité d'empaqueter le matériel didactique (*hérité de l'IMS*), et des méta-données pour l'importation et l'exportation (*développé par l'IMS Global Learning Consortium et l'IEEE*) ont été ajoutées. Le paquet content est un morceau intégral pour la réunion une des conditions globales du SCORM - « Interopérabilité ».

3) SCORM 2004 (*Janvier 2004*)

Avec la publication de SCORM 2004, ADL a décidé de modifier le versionnage de SCORM de façon à ce que chaque document puisse être mis à jour de façon indépendante. Le nombre de spécifications et la dimension propre des documents ont rendu ce changement nécessaire pour assurer la gestion des révisions et des corrections apportées à l'ensemble des documents. Chaque document SCORM comprend maintenant sa version propre qui commence par la présente publication (*aux fins historiques*) de la « Version 1.3 ». À l'avenir, les changements ne s'appliqueront qu'au document visé et n'affecteront que le numéro de version de ce document. Aujourd'hui on a trois éditions du modèle SCORM 2004.

Le modèle SCORM 2004 (édition 2^{ème} – juillet 2004) est le modèle le plus stable ([2], p.9), qui va être utilisé dans mon travail.

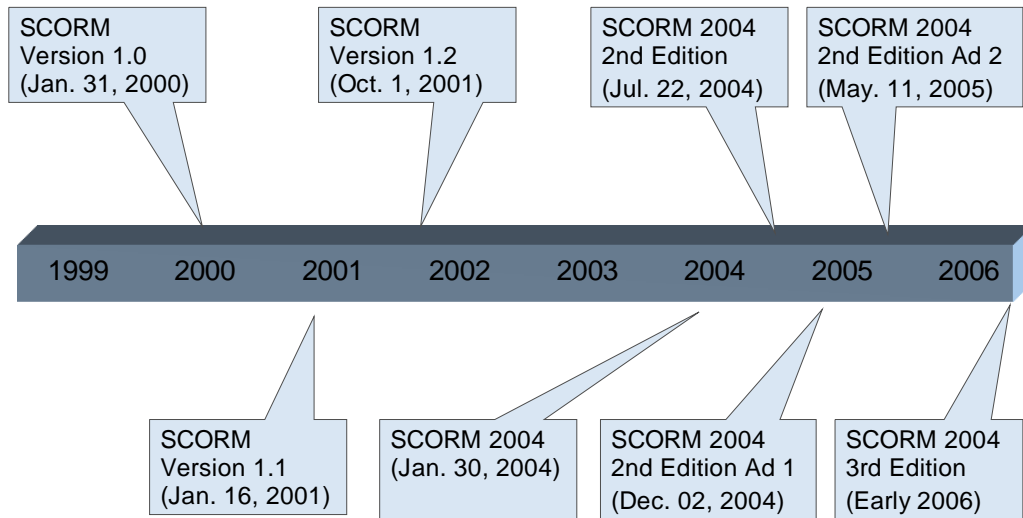


Figure 2.2 : Chronologie du SCORM

2.2. Modèle SCORM 2004

2.2.1. Introduction au modèle SCORM 2004

Le modèle SCORM 2004 (édition 2^{ème} – juillet 2004) se compose de trois composants principaux avec plus de 2000 pages de 4 documents ([9], p.37-42):

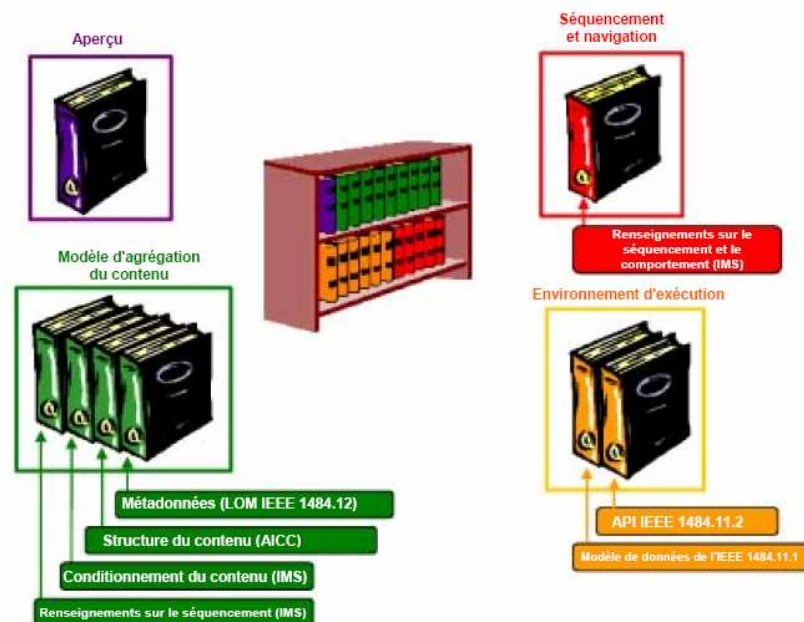


Figure 2.3 : L'organisation du modèle SCORM 2004

(1) Le Modèle d'agrégation du contenu (*Content Aggregation Model - CAM*) : c'est le composant très important de ce modèle permettant d'organiser le contenu d'apprentissage à partir des ressources d'apprentissage ([12], p.13).

(2) L'environnement d'exécution (*Run-Time Environment – RTE*) : C'est un composant du modèle SCORM permettant aux LMS de chercher les SCOs et de communiquer avec eux en temps d'exécution basé les requêtes d'apprenant et les règles de séquençement ([2], p.13 et [14], p.17).

(3) Le séquençement et la navigation du SCORM (*The SCORM Sequencing and Navigation – SN*) : Hériter des spécifications de *IMS Simple Sequencing (SS)* pour définir comment les activités d'apprentissage dans un cours peuvent être livrées basées l'interaction d'utilisateurs avec le cours ([16], p.20, 23).

2.2.2. Modèle d'agrégation du contenu (CAM)

Le Modèle d'agrégation du contenu CAM dispose les règles et les mécanismes qui sont utilisés pour associer des ressources d'apprentissage (les fichiers individuels) à un SCO (Sharable Content Object) et des SCOs à une Organisation de contenu. Il consiste en trois sous-composants ([12], p.14) comme suivants:

1) Le Modèle du contenu (*SCORM Content Model - CM*): Décrire les composants du modèle SCORM qui rendent les ressources d'apprentissages partageables qui constituent une expérience d'apprentissage et expliquer comment ces ressources peuvent être associés ([2], p.11 et [12], p.21). Chaque ressource d'apprentissage **le plus petit** est comprise comme une représentation électronique des médias, tels que des textes, des images, des sons, des objets d'évaluation ou n'importe quel morceau de données qui peuvent être rendues par un web client et être affichées aux apprenants ([12], p.21). On les appelle : **les Assets**.

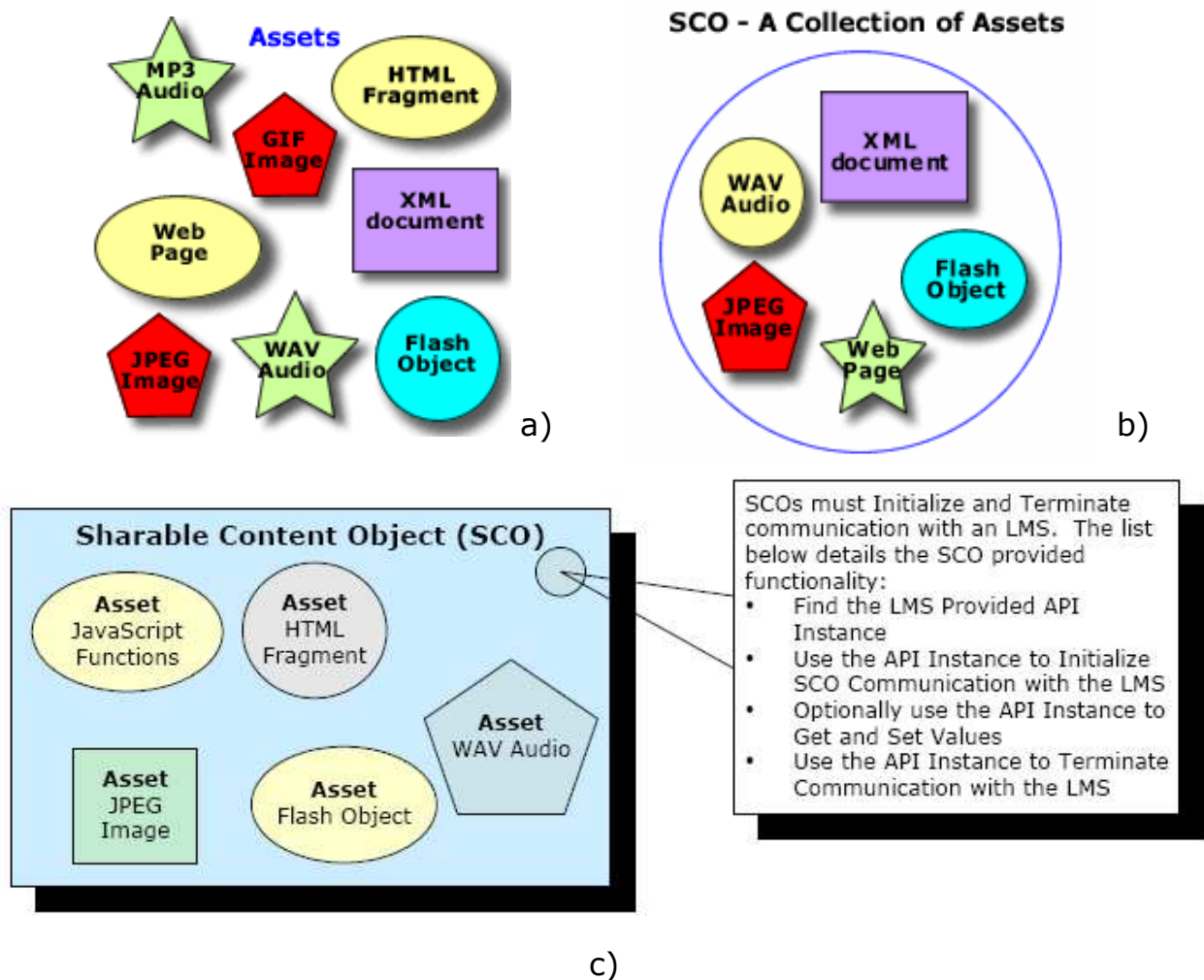


Figure 2.4 : Des Assets (a) et des SCO (b et c)

Un ou plusieurs **Assets** peuvent être associés pour former un objet de contenu partageable ou on l'appelle **SCO** (*Sharable Content Object*). Chaque **SCO** est compris comme une unité de contenu d'apprentissage indépendante. Il peut communiquer avec des plates-formes (*LMS – Learning Management System*) grâce aux APIs (*Application Programming Interface – voir figure 2.4 c*) en utilisant le modèle SCORM RTE (*Run-Time Environment*) et il peut être géré par le LMS ([12], p.21-23).

Une collection des SCOs qui est arrangée selon un arbre d'activité d'apprentissage (voir figure 2.5) peut être associée dans une Organisation de contenu (*Content Organization – CO*) ([12], p.23-24). Une CO peut être

comprise comme un plan représentant les ressources d'apprentissage correspondant aux taxonomies différentes du contenu d'apprentissage et elle peut être une portion d'un cours, un cours complète ou une collection de cours.

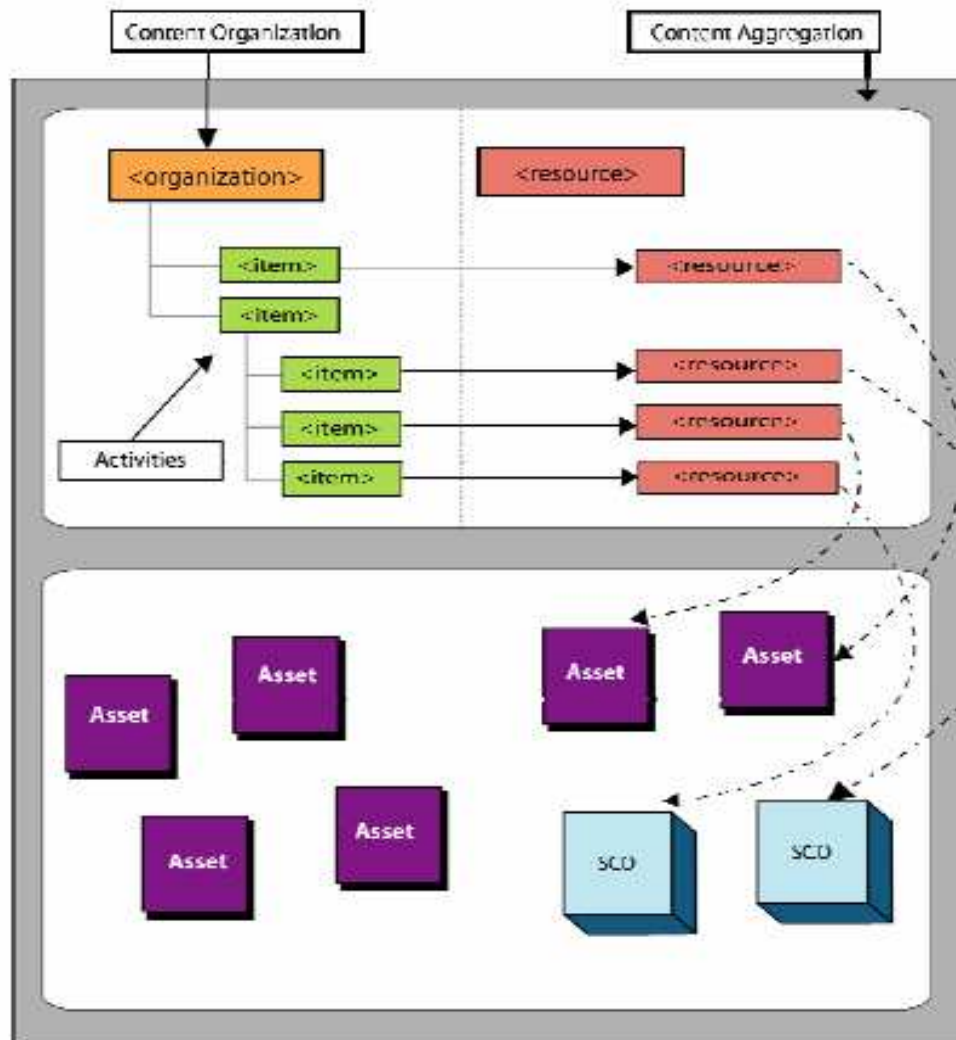


Figure 2.5 : Organisation de contenu

Ici chaque activité dans une organisation de contenu peut mettre en référence des méta-données pour tenir compte de la recherche et de la découverte dans des dépôts, permettant de ce fait des occasions pour la réutilisation. La méthode d'associer des activités aux méta-données est le paquet de contenu (voir section 2.2.2 - 2).

Toutes les unités de contenu ci-dessus (Assets, SCOs et COs) sont des composants essentiels du modèle de contenu du CAM pour pouvoir réaliser une expérience d'apprentissage.

2) Conditionnement du contenu (SCORM Content Packages - CP) : Pour faire faciliter l'échange de contenu d'apprentissage entre un développeur de contenu et LMS, on utilise un Conditionnement du contenu (*Content Package*) grâce à la notion de **Manifest** qui ont des informations nécessaires (présenté dans le format XML) pour livrer le contenu qui renferme dedans ([12], p.29-41). En général un CP peut présenter pour un cours complète. Un CP consiste en deux composants essentiels qui est tassés dans un fichier *.ZIP et facile à distribuer sur Internet:

a) Le fichier XML qui s'appelle **Manifest** : comme IMS Manifest qui décrits tous les contenus, leur organisation et tous les ressources renfermés dans ce CP.

b) Les fichiers de contenu d'apprentissage (Assets et SCOs).

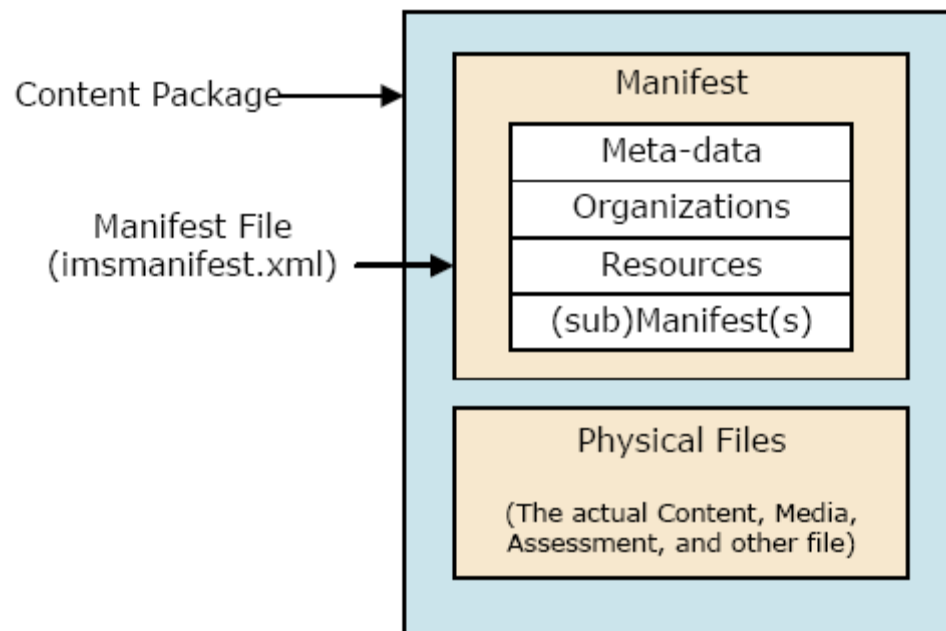


Figure 2.6 : Conditionnement du contenu (SCORM Content Packages)

3) Méta-données (SCORM Meta-Data) : Le CAM dispose aussi des spécifications décrivant pour les unités de contenu (COs, SCOs ou Assets). Elles sont des Méta-données. Ces informations rendent les Assets et les SCOs **cherchable**, **réutilisable** et **découvrable** par des LMS et permettent aux apprenants de trouver vraiment le contenu d'apprentissage qu'ils veulent ([12], p.93-200 et [2], p.10).

2.2.3. Environnement d'exécution (RTE)

Quand LMS reçoit une requête d'un apprenant, il cherche le SCO correspondant et le livre à l'apprenant. Ce SCO va utiliser ECMAScript/JavaScript pour chercher et appeler les fonctions de l'API (*Initialize*, *Terminate*, *GetValue*, *SetValue* et *Commit*) qui sont disponible dans LMS, et qui les aident à communiquer avec des LMS différents ([2], p.14 et [14], p.8, 26-29, 33-42, 60-64).

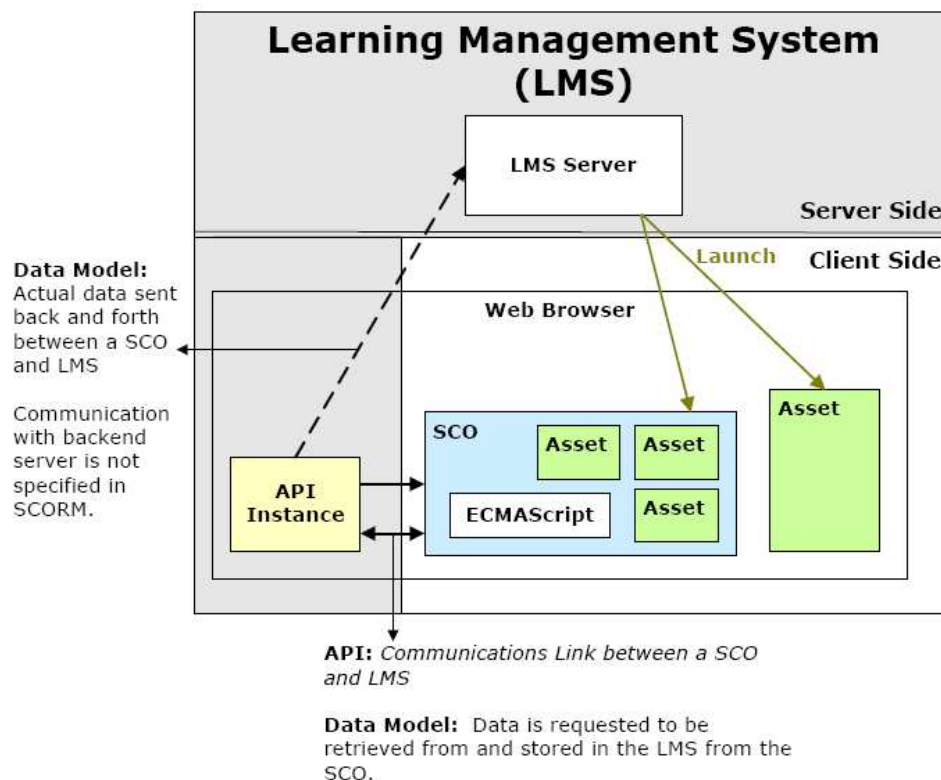


Figure 2.7 : L'environnement d'exécution du modèle SCORM (RTE)

1) Modèle de données de SCORM RTE

Cette communication base sur les données de *SCORM RTE Data Model* qui sont une collection de données utiles (22 types de données) pour suivre les cours (information sur les apprenants, les interactions entre l'apprenant et ce SCO, les états complets d'apprentissage, information sur objectifs d'apprentissage...) ([2], p.14-16; [14], p.67-194). Toutes ces données sont préfacées avec « CMI » et on l'appelle « CMI » modèle (*Computer Managed Instruction*) qui est hérité de Data Model Standard de AICC. En utilisant ces données, les SCOs peuvent communiquer avec des LMS différents. Par exemple on utilise **cmi.location** pour que le SCO puisse marquer le lieu auquel l'apprenant peut retourner... Pour pouvoir « parler » avec un LMS, un SCO utilise les **huit fonctions de l'API** qui sont disponibles dans tous les LMS conformant la norme de SCORM.

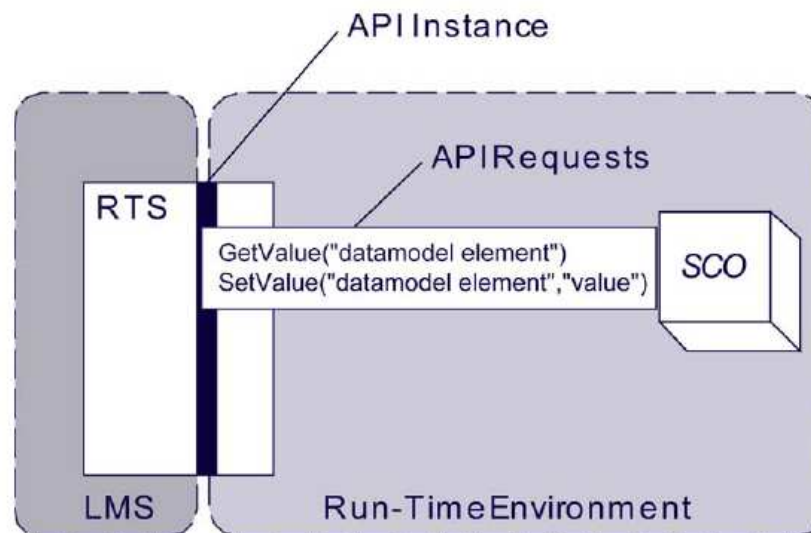


Figure 2.8 : Utilisation du Modèle de données avec l'objet d'API

2) Fonctions de l'API: Ce sont les huit fonctions suivantes :

a) Initialize(): Cette fonction est utilisée pour amorcer la communication entre SCO et LMS. Un SCO ne peut « parler » avec LMS qu'il a réussi dans la phase initiale.

b) `Terminate()` : Cette fonction est utilisée pour terminer la communication entre SCO et LMS.

c) `GetValue()` : Cette fonction est utilisée pour prendre des données nécessaires qui sont sauvegardées au LMS.

d) `SetValue()` : Cette fonction est utilisée pour permettre au SCO de sauvegarder au tampon de LMS des données nécessaires.

e) `Commit()` : Cette fonction est utilisée pour sauvegarder toutes les données qui sont disponibles dans le tampon de LMS dans le modèle CMI. Et les trois fonctions `GetLastError()`, `GetErrorString()` et `GetDiagnostic()` pour déterminer les erreurs possibles en communiquant.

2.2.4. Séquencement et la navigation du SCORM (SN)

Le document SCORM sur le SN décrit comment le contenu conforme au SCORM peut être ordonné au moyen d'événements de navigation lancés par l'apprenant ou par le système.

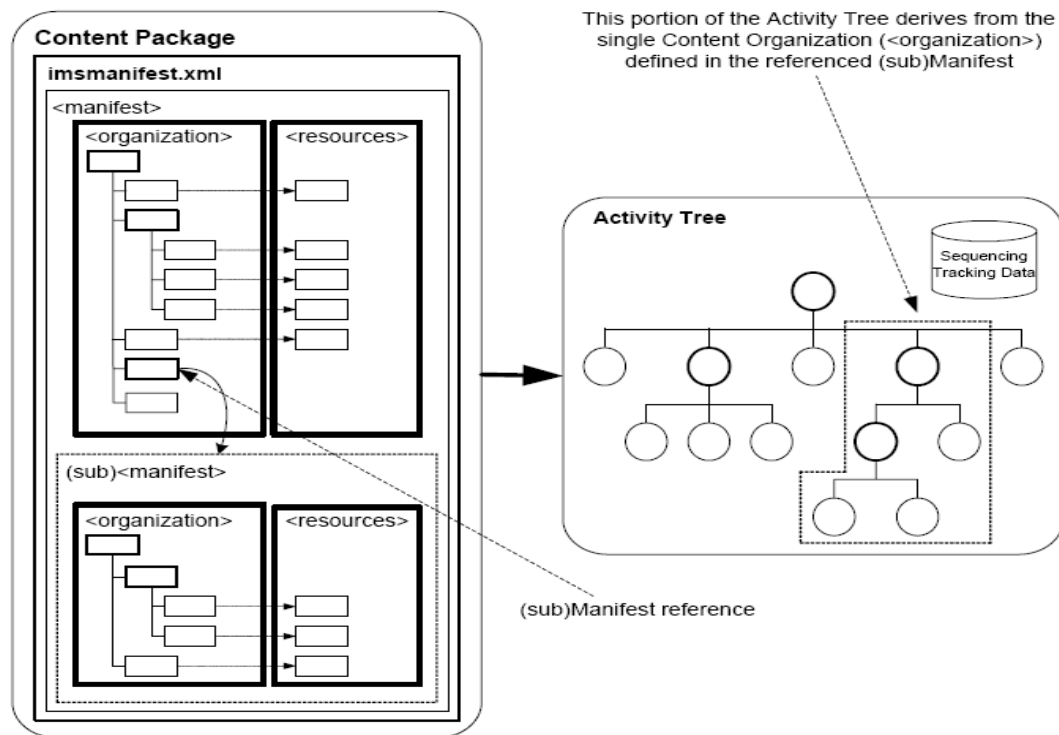


Figure 2.9 : Relation entre un CO et un arborescence d'activité

Le modèle de SN du SCORM décrit les comportements et les fonctionnalités que les LMS conformes au SCORM doivent mettre en oeuvre pour assurer le traitement des renseignements sur le séquençement lors de l'exécution. De façon plus précise, il décrit les ramifications et la progression des activités d'apprentissage **sous forme d'une arborescence des activités**, laquelle est fondée sur les résultats des interactions d'un apprenant avec les objets de contenu et une stratégie de séquençement consignée.

Dans les processus d'apprentissage LMS va traduire l'Organisation de contenu (CO) dans une arborescence des activités correspondante.

1) Quelques notions correspondantes

a) **L'arborescence d'activité** : L'arborescence des activités est une structure conceptuelle des activités d'apprentissage gérées par le LMS pour chacun des apprenants, comme l'illustre la figure 2.10. Dans le modèle SCORM, une activité d'apprentissage peut faire allusion à des objets de contenu qui sont dispensés à l'apprenant.

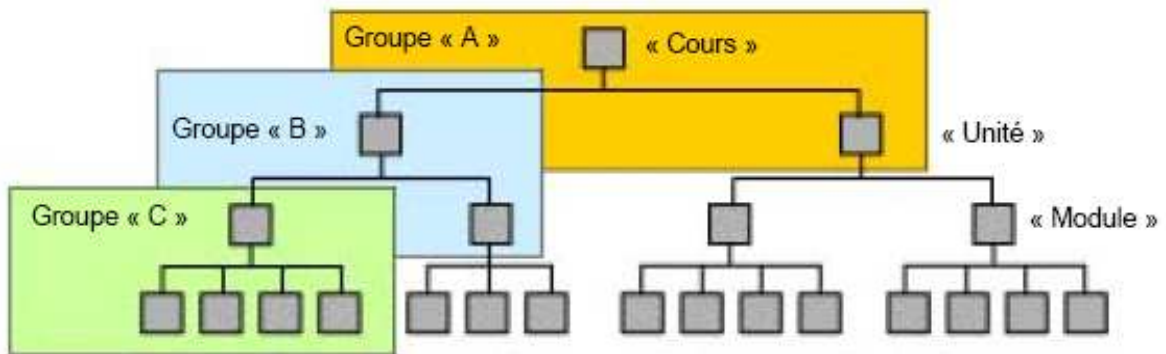


Figure 2.10 : Arborescence et groupes conceptuels liés aux activités

Chaque activité a un état de la piste associée à chaque étudiant qui est assigné à l'activité. Le comportement du LMS est décrit en termes de traverser les nœuds de l'arborescence pour déterminer quelle activité à livrer à l'apprenant ([2], p.16, 23-33).

b) **Groupe d'activité** : C'est une activité d'apprentissage qui a des sous-activités. Un cours peut être compris comme un groupe d'activité.

c) **Un Essai** (*an Attempt*) : Chaque fois un apprenant essaie de compléter une activité on dit qu'il a fait un essai.

d) **Objectif d'apprentissage** : décrits l'état complète d'une activité d'apprentissage. Une activité a un ou plusieurs objectifs associés avec elle.

Quand un apprenant choisi de interagir avec les contenus représentés par l'arborescence, LMS évalue l'enchaînement et suis à la piste pour déterminer la séquence relative d'activité d'apprentissage.

Chaque fois un apprenant essaie d'exécuter une activité d'apprentissage (un Essai) et la complète, il satisfait un objectif d'apprentissage. Pendant le temps d'un Essai, on peut aussi suspendre cette activité et la reprendre après.

2) Modes de contrôle du séquencement (Sequencing Control Modes-SCM)

Un ensemble de spécifications permettant aux apprenants de visiter les sous-activités dans un groupe est défini dans SCM. Il se compose de :

+) **Sequence Control Choice** : Permet de choisir librement des sous-activités dans un Cluster ou non.

+) **Sequence Control Choice Exit** : Permet de visiter des activités dehors le Cluster ou non.

+) **Sequence Control Flow** : Permet de déplacer entre parents-activités et sous-activités par « Continue » et « Previous » ou non.

+) **Sequence Control Forward Only** : Permet de déplacer seulement de parents-activités à sous-activités.

3) Règles de séquencement (Sequencing Rules - SR)

Chaque activité peut avoir zéro ou plusieurs SR. Ces règles sont évaluées à ce moment spécifié. Basé sur un ensemble de condition qui sont

TRUE, certaines actions vont être exécutées. Les règles de séquençement permettent aux apprenants de pouvoir avoir une expérience d'apprentissage dynamique et personnelle basant sur leur progrès et succès durant leur complétion des activités et objectifs.

Conditions pour ces règles peuvent être la satisfaction des objectifs d'apprentissage, de l'état complète d'une activité... Les types d'action possibles sont :

- **Pre-Condition Actions** : sont utilisées quand on parcourt une arborescence d'activité pour déterminer l'activité suivante à l'apprenant. Il y a des actions ci-dessous :

+) **Skip** : L'activité n'est pas considérée comme un candidat pour la délivrer à l'apprenant.

+) **Disabled** : L'activité est invalide si la condition est satisfaite.

+) **Hidden from Choice** : L'activité n'est pas choisie par l'apprenant.

+) **Stop Forward Traversal** : L'activité va prévenir des activités suivantes quand elle est choisie par l'apprenant.

- **Post-Condition Actions** : sont utilisées quand on termine un Essai sur une activité. Il y a des actions suivantes :

+) **Exit Parent** : La terminaison de l'activité-père est traitée.

+) **Exit All** : La terminaison de toutes les activités est traitée.

+) **Retry** : Essayer encore une fois l'activité.

+) **Continue** : Essayer sur l'activité suivante dans l'ordre prédéfinie.

+) **Previous** : Retourner l'activité précédente dans l'ordre prédéfinie.

- **Exit Action** : est utilisé quand on termine un Essai sur une activité descendante.

Chapitre 3

Processus de transformation d'un cours en ligne à la norme de SCORM 2004

Dans ce chapitre nous allons proposer les processus en général de transformation d'un cours en ligne à la norme de SCORM 2004. On a les trois étapes principales suivantes :

3.1. Etape préparatoire

Objectif : Obtention des contenus d'apprentissage sur la machine locale et édition de ces contenus bruts s'il est nécessaire.

L'entrée : Les contenus d'apprentissage disponible en ligne.

La sortie : Les contenus d'apprentissage pour le cours étant disponible sur la machine locale.

Avec cette étape on peut utiliser quelques outils libres existants pour télécharger les contenus d'apprentissage qui sont disponible sur un site web extérieur à la machine locale : **HTTrack** (<http://www.httrack.com>), Plugin pour FireFox **Downthemall** (<http://www.downthemall.net>), Plugin pour FireFox **FlashGot** (<http://www.flashgot.net>)...

Tous ces outils ci-dessus permettent aux utilisateurs de télécharger un Site Web entier à la machine locale. Mais avec des Site Web ayant la restriction d'accès, on doit d'abord contacter l'administrateur pour avoir un permis d'accès au Site Web afin de télécharger tous les contenus d'apprentissage dont on a besoins.

La plupart de cas, les contenus téléchargés consistent en quelques pages inutiles. On doit, donc, rééditer ces contenus, supprimer des pages

dont on n'a besoins pas, ajouter des pages s'il est nécessaire à la main afin d'avoir vraiment le contenus du cours.

3.2. Etape de transformation d'un cours local à la norme de SCORM 2004

Objectif : Transformation d'un cours étant disponible sur la machine locale à la norme de SCORM 2004.

L'entrée : Les contenus d'apprentissage (fichiers *.html) qui sont disponibles sur la machine locale.

La sortie : Un paquet de contenus en norme de SCORM 2004 qui va être importé sur une plate-forme choisie.

3.2.1. Etude de la structure d'un cours en norme de SCORM 2004

En général un cours peut être divisé en plusieurs unités ou chapitres (voir la section 2.2.4). Chaque unité ensuite se compose de plusieurs modules ou leçons. Une leçon est un ensemble des morceaux de contenus d'apprentissage en ordre comme textes, images, schémas, vidéo... Au point du professeur, une leçon c'est une unité de contenu d'apprentissage la plus petite qu'il peut gérer.

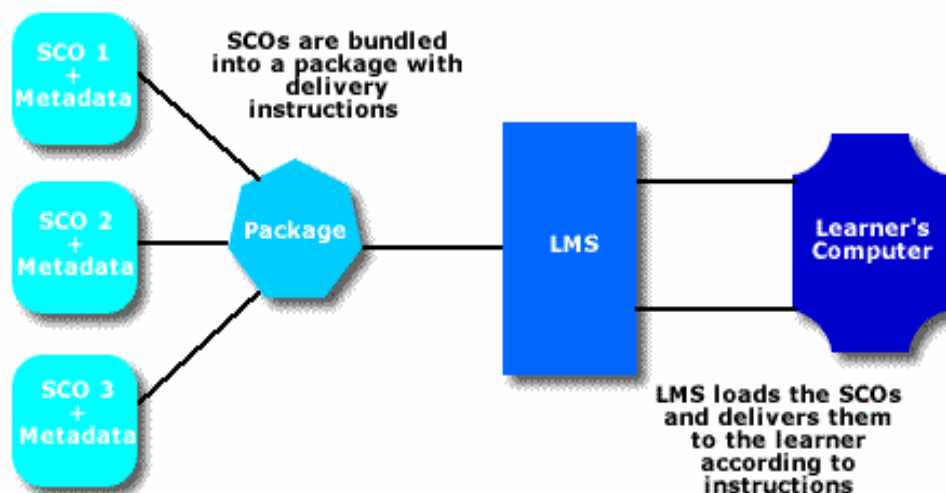


Figure 3.1 : Modèle d'utilisation des SCO

En norme de SCORM 2004 donc, en général, une leçon correspondante à un SCO (une activité) qui est géré par un LMS. Chaque chapitre est un ensemble des SCO en ordre prédéfinis qui est correspondant à une Organisation de contenu (CO). Et un cours est un ensemble des CO en ordre prédéfinis qui est correspondant à un Paquet de contenu (CP) (correspondant à une arborescence des SCO). **Donc, pour transformer un cours local à la norme de SCORM on a deux étapes principales :**

- 1) Transformation des fichiers du cours aux SCO correspondants en norme de SCORM 2004.
- 2) Construction d'un Paquet de SCORM 2004 à partir des SCO créés.

3.2.2. Etudes des étapes de transformation d'un cours local à la norme de SCORM 2004

1. Etude de la transformation des fichiers du cours aux SCO

Objectif : Transformation des fichiers du cours aux SCO en norme de SCORM 2004.

L'entrée : Les fichiers d'apprentissage *.html qui sont disponibles sur la machine locale.

La sortie : Des SCO en norme de SCORM 2004.

a) Première étape : Etude l'architecture d'un SCO.

Un SCO se compose d'un ou plusieurs pages *.html en ordre prédéfinis et il peut communiquer avec LMS via l'objet d'API pour obtenir ou sauvegarder des informations nécessaires sur des Essais des apprenants, sur l'état complète de l'activité, sur les apprenants... (Voir le chapitre 2).

Problème 1: L'interface de communication entre LMS et les SCO est limité (on n'a que huit fonctions) mais les fonctions qui sont disponibles dans les SCO sont diverses et dépendantes de chaque SCO.

Solution : Division les fonctions de chaque SCO en deux couches :

- **Couche de système** : s’occupe de chercher l’objet d’API et crée les fonctions d’interface pour la couche d’application (voir rubrique c suivante) à partir de huit fonctions de base de LMS.

- **Couche d’application** : s’occupe de contrôler chaque SCO.

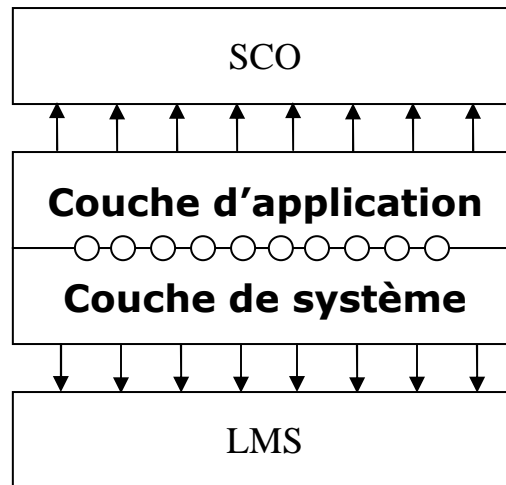


Figure 3.2 : L’architecture en deux couches de SCO

Problème 2 : Dans chaque SCO doit utiliser toutes les fonctions nécessaires dans les deux couches. Une question qui se pose : Comment on peut organiser les fonctions dans deux couches pour les réutiliser dans chaque SCO ?

Solution : On organise chaque couche dans une librairie (un fichier) et chaque SCO va les appeler via l’objet d’API.

b) Deuxième étape : Trier les fichiers en ordre.

L’entrée : Les fichiers d’apprentissage *.html qui sont disponibles sur la machine locale.

La sortie : Les fichiers d’apprentissage en ordre prédéfinis par le professeur.

Cette étape on doit faire à la main, parce qu’on n’a pas encore un outil qui peut remplacer le professeur (c’est trop difficile).

c) Troisième étape : Construction des librairies et Insertion de ces fonctions aux SCO.

L'entrée : Les fichiers d'apprentissage en ordre prédéfinis par le professeur.

La sortie : Les deux librairies en deux couches et des SCO avec des Scripts de communication.

Pour qu'un SCO puisse communiquer avec LMS, il faut fournir un objet d'API à côté du LMS qui permet au SCO d'appeler ses fonctions via **ECMAScript** et les exécuter. En SCORM 2004, cet objet s'appelle : **API_1484_11**. Donc, chaque SCO qui veut communiquer avec LMS doit chercher et trouver cet objet et l'utilise pour appeler ses fonctions.

On peut voir les transactions nécessaires pour qu'un SCO puisse communiquer avec un LMS conformant SCORM 2004 sur la figure 3.4. Il y a trois phases principales :

- **Phase initiale** : SCO cherche l'objet d'API et l'utilise pour appeler la fonction d'initiation : `Initialize()`. On peut utiliser aussi les trois fonctions : `GetLastError()`, `GetErrorString()` et `GetDiagnostic()` pour savoir l'état d'Initiation.

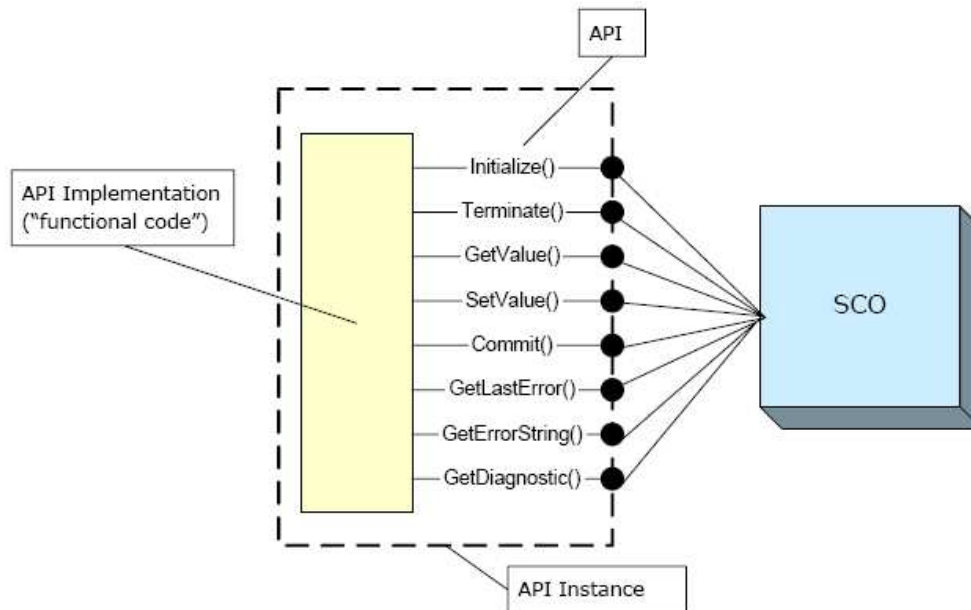


Figure 3.3 : API fonctions, Objet d'API et Implémentation d'API

- **Phase de communication** : Après d'Initiation, chaque SCO peut utiliser les cinq fonctions : GetValue() et SetValue() pour obtenir ou sauvegarder les informations nécessaires ; et GetLastError(), GetErrorString() et GetDiagnostic() pour savoir l'état de la communication.

- **Phase de terminaison** : Après de la phase de communication avec LMS, le SCO peut utiliser la fonction Terminate() pour terminer la communication. Dans cette phase là on peut aussi utiliser les trois fonctions : GetLastError(), GetErrorString() et GetDiagnostic() pour savoir l'état de la terminaison de la communication.

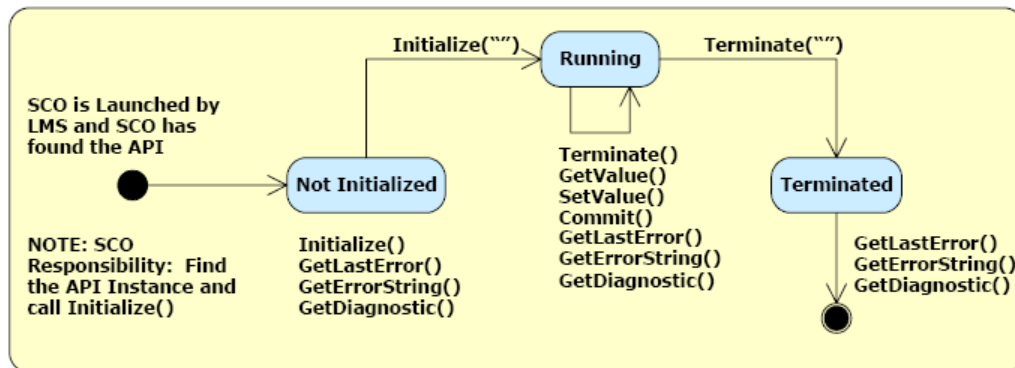


Figure 3.4 : Des transactions de communication entre LMS et SCO

Donc, en couche de système on a des fonctions suivantes (voir l'**Annexe A** pour savoir en détail) :

- getAPIHandle() : Pour trouver l'objet d'API.
- initializeCommunication() : Pour amorcer la communication.
- terminateCommunication() : Pour terminer la communication.
- retrieveDataValue(name) : Pour obtenir une information du LMS.
- storeDataValue(name, value) : Pour sauvegarder une information Au LMS.
- retrieveLastErrorCode() : Pour avoir des erreurs dans la communications
- retrieveErrorInfo(errCode) : Pour savoir la description des erreurs.

- retrieveDiagnosticInfo(error) : Aider les développeur du contenu de trouver des erreurs dans un SCO.
- fonction persistData() : demande au LMS de sauvegarder toutes les informations dans le tampon.
- displayErrorInfo(errCode) : afficher l'erreur à l'écran.

Avec la couche d'application le nombre des fonctions dépend de chaque activité d'apprentissage (SCO). Il y a de 22 types de données avec environs de 50 éléments dans le modèle de données du SCORM RTE pour tous les SCO possibles (voir l'Annexe B et chapitre 2 pour savoir en détail). Dans ce travail on utilise les éléments de données suivantes :

- **cmi.location** : Pour marquer l'emplacement où l'apprenant sort dans l'Essai (Attempt) précédant (comme un "bookmark").

- **cmi.entry** : Cet élément contient l'information qui affirme que l'étudiant a précédemment consulté le SCO. Il y a des valeurs ci-dessous :

- +) **ab_initio** : L'apprenant n'a pas Essai sur le SCO.

- +) **resume** : L'apprenant avait Essai sur le SCO, mais il ne le complète pas. Il exit le SCO avec état « suspend ».

- +) **_nil_** : On ne peut pas savoir exactement l'état de cet élément.

- **cmi.completion_status** : Indication l'état de la complétion de ce SCO.

Il y a des valeurs comme suivantes :

- +) **completed** : L'apprenant a complété ce SCO.

- +) **incomplete** : L'apprenant n'a pas encore complété ce SCO.

- +) **not_attempted** : L'apprenant n'a pas Essai sur ce SCO.

- +) **unknown** : On ne peut pas affirmer l'état de la complétion de ce SCO.

- **cmi.exit** : Déterminer comment et pourquoi l'apprenant sort du SCO. Il y a des values comme suivantes :

- +) **timeout** : Le SCO est terminé à cause de l'excédent du temps qu'on a précisé pour lui.

+) **suspend** : L'apprenant quitte ce SCO sans le compléter et il veut le essayer de le compléter plus tard.

+) **logout** : L'apprenant quitte l'activité d'apprentissage où ce SCO est un composant.

+) **normal** : L'apprenant quitte le SCO normalement.

+) **nil** : On ne peut pas déterminer l'état de la terminaison du SCO.

- **cmi.learner_name** : LMS va utiliser cet élément pour gérer les apprenants. Il sauvegarde le nom de l'apprenant à côté du serveur.

- **cmi.learner_Id** : LMS utilise cet élément pour identifier des apprenants qui ont essayés sur le SCO.

- **cmi.comments_from_learner** : Cet élément permet aux apprenants de commenter ce SCO quand ils le essaient. Il y a les quatre sous éléments ci-dessous :

+) **cmi.comments_from_learner._count** : Pour déterminer le nombre des commentaires sur ce SCO.

+) **cmi.comments_from_learner.i.location** : Pour déterminer la position où le commentaire i_ième est fait.

+) **cmi.comments_from_learner.i.comment** : Pour sauvegarder le commentaire i_ième des apprenants sur ce SCO au serveur LMS.

+) **cmi.comments_from_learner.i.timestamp** : Pour déterminer le temps où l'apprenant a fait commentaire i_ième sur ce SCO.

On utilise tous les éléments de données SCORM RTE ci-dessus pour construire les fonctions nécessaires dans la librairie de couche d'application (voir l'Annexe A).

Insertion des fonctions construites aux SCO : Pour qu'un SCO puisse communiquer avec LMS on doit insérer les fonctions JavaScripts dans les fichiers *.html qui construit ce SCO. Ce travail est très lourd et il doit être réalisé par des développeurs de contenu qui sont programmeurs aussi.

Problème 3 : Comment on peut insérer **automatique** des fonctions JavaScripts aux SCO ?

La solution de ce problème est présentée en détail dans le chapitre 4.

2. Etude de la construction d'un Paquet de SCORM 2004

Objectif : Construction d'un Paquet de SCORM 2004 à partir des SCO.

L'entrée : Les SCO du cours.

La sortie : Un Paquet de contenu en norme de SCORM 2004 de ce cours (un CP).

Pour créer un CP à partir des SCO disponibles on doit établir un fichier nommé `imsmanifest.xml` qui contient des informations nécessaires pour associer les fichiers physiques et les SCO dans ce CP (ce cours).

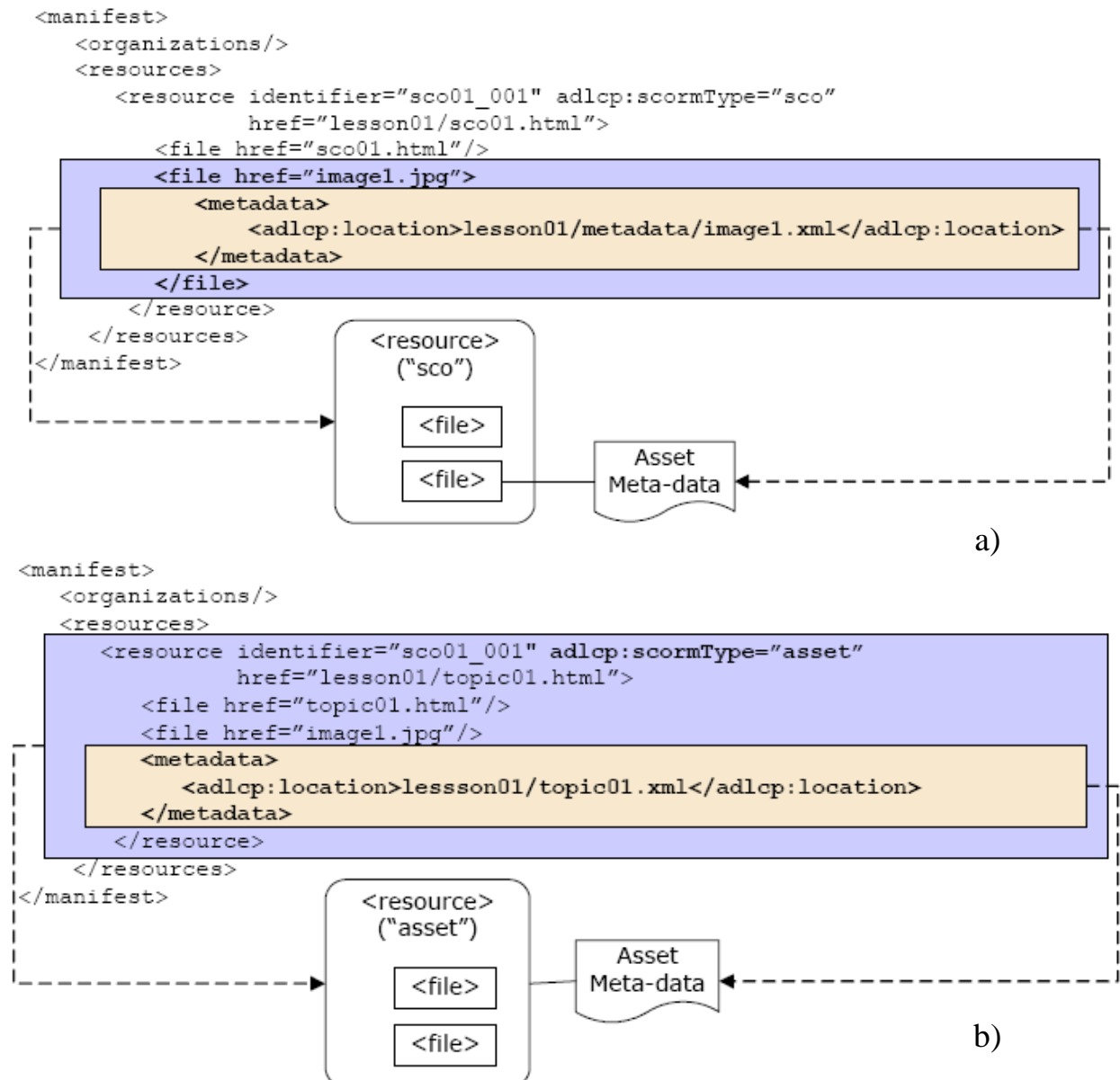


Figure 3.5 : La structure du fichier `imsmanifest.xml` pour associer un Asset

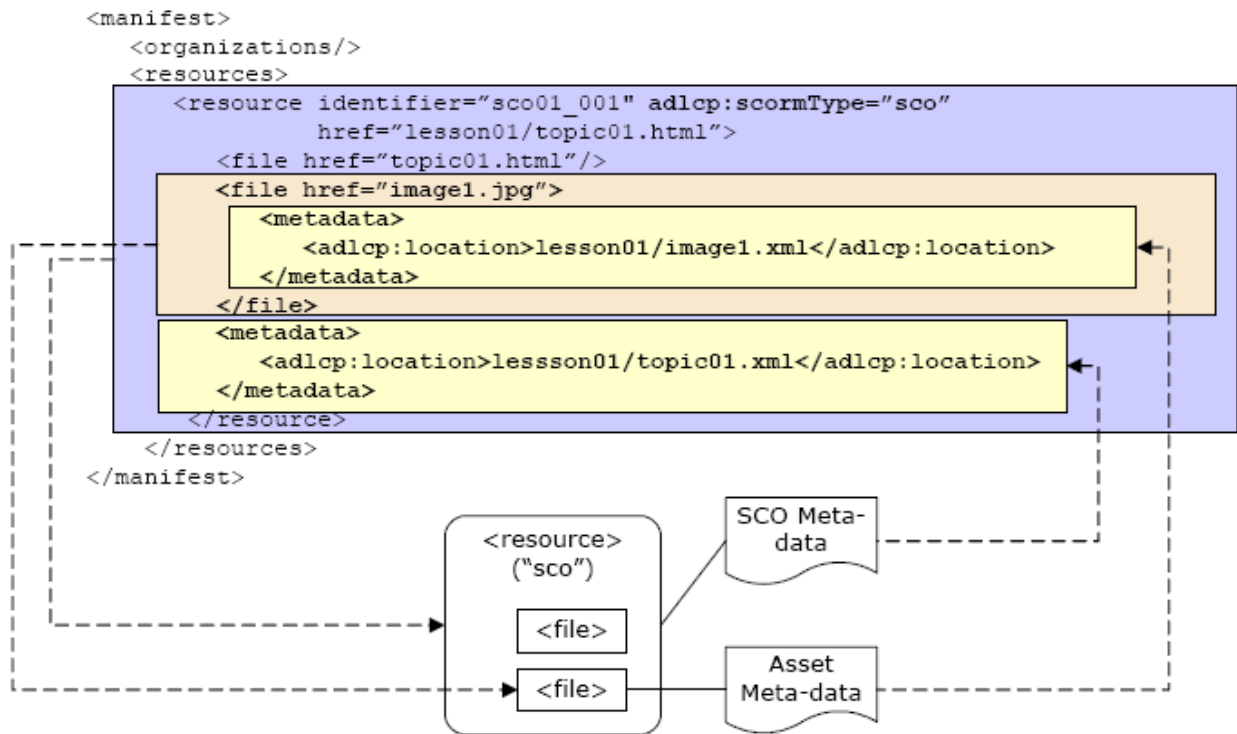


Figure 3.6 : La structure du fichier imsmanifest.xml pour associer un SCO

Ce travail est très lourd et il doit être réalisé par des développeurs de contenu qui sont programmeurs et un spécialiste en norme de SCORM.

Problème 4 : Est-ce qu'un professeur sans connaissances de SCORM et de programmation peut créer un CP ?

Solution : On utilise un des outils « open sources » qui sont disponibles sur Internet et qui permet aux professeurs de transformer des SCO en paquet.

Aujourd'hui il y a plusieurs outils « open sources » de transformation des SCO en paquet comme : Reload Editeur, FCK Editeur, EXE Editeur... Dans ce travail nous choisissons Reload Editeur (voir chapitre 4).

3.3. Etape d'importation sur une plate-forme choisie

Objectif : Choisir une plate-forme et l'importation d'un paquet de contenus en norme de SCORM 2004 sur cette plate-forme choisie.

L'entrée : Un paquet de contenus d'apprentissage en norme de SCORM 2004 et une plate-forme choisie.

La sortie : Un cours en ligne en norme de SCORM 2004 sur la plate-forme choisie.

3.3.1. Choix d'une plate-forme

Maintenant on a environs plus de 235 plates-formes différentes (voyez chapitre 1). Il est, donc, difficile de choisir une plate-forme. Je propose une façon de choisir comme suivante :

- 1) D'abord on choisit quelques plates-formes les plus utilisées.
- 2) Donner quelques critères importants pour en choisir une.

Selon Richard Wyles ([25], p.9) on utilise souvent dans le monde entier les plates-formes ci-dessous :

- **Sakai** : SAKAI est une initiative en code source libre initiée par quatre universités américaines (University of Michigan, Indiana University, Massachusetts Institute of Technology et Stanford University) qui a pour objectif de développer un environnement collaboratif d'apprentissage (ENA). Dans sa première incarnation, SAKAI intégrait les outils antérieurement développés par les universités constituantes au moyen d'une réingénierie assurant la portabilité et l'interopérabilité. L'initiative SAKAI s'appuie, par ailleurs, sur les bases technologiques et fonctionnelles mises de l'avant par le consortium uPORTAL et par l'Open Knowledge Initiative (OKI) ([26], p.19).

- **Concept@** : Le portail Concept@, dont le développement est très avancé (une version est expérimentée à la TÉLUQ), fait suite aux travaux que la TÉLUQ entreprenait en 2003 afin d'intégrer les fonctionnalités des trois différents ENA utilisés au sein de l'établissement (Adap@Web, Explor@ et Sydife). La démarche est motivée, à la fois par la volonté interne de construction du Campus Virtuel de la TÉLUQ, par l'importance nouvelle que prend la formation en ligne dans le cadre du rattachement de l'établissement à l'UQAM, par la création du Conseil de la formation à distance (CQFD) au sein du réseau de l'Université du Québec et par le mouvement international qui se

déploie autour des normes et des standards d'interopérabilité et autour de la constitution de réseaux de référentiels d'objets pédagogiques ([26], p.17).

- **Moodle** : Moodle est une plate-forme disponible en code source libre qui a été développée en Australie. Elle résulte de l'effort de développement d'un ancien de WebCT, Martin Dougiamas, qui insatisfait par sa structure de fonctionnement a décidé de produire une plate-forme reproduisant les fonctionnalités de WebCT tout en les améliorant. Le développement en code source libre permettait, par ailleurs, une plus grande flexibilité d'adaptation et la possibilité de collaboration. Moodle a été un succès puisque la communauté des utilisateurs est importante, représentant plus de 50 000 utilisateurs répartis dans 160 pays. Ceux-ci participent au développement de l'application ([26], p.18).

- **UQTR** : Le Portail de cours de l'UQTR crée automatiquement, pour tous les cours offerts par l'UQTR, des sites Web intégrant des outils pédagogiques et administratifs conviviaux. Il offre un accès personnalisé et sécurisé aux enseignants et aux étudiants. Plusieurs outils pédagogiques sont regroupés dans le Portail de cours, ce qui favorise une pédagogie active tout en limitant les transactions administratives ([26], p.16).

Pour pouvoir choisir une des 4 plates-formes ci-dessus on peut utiliser des critères (tableau 3.2). En utilisant des poids pour chaque critère on a les résultats (Tableau 3.1 source [26], p. 21):

Critères généraux obligatoires				
	Concept@	Sakai	Moodle	UQTR
Flexibilité et décentralisation	1	2	2	0
Architecture de base	0	2	2	0
État de la communauté	0	1	2	0
Support commercial	0	1	1	0
Support multilingue	1	1	2	0
Standards	1	1	1	0
Support et documentation	0	1	2	2
Base de données	2	2	2	1
Modularité et adaptation	1	2	2	1
Gestion de la croissance	2	2	2	1
Sécurité	1	2	1	1
Robustesse et fiabilité	1	2	2	1
Accès par fureteur	2	2	2	2
Total	12	21	23	9

On peut voir ici c'est Moodle qui est la meilleure plate-forme. On choisit, donc cette plate-forme dans ce travail.


Tableau 3.2 Description des critères pour choisir une plate-forme (source [26], p.32) :

Description des critères généraux obligatoires	
Flexibilité et décentralisation	Permettre sans coûts exorbitants d'avoir des installations personnalisées selon les besoins des facultés tout en bénéficiant d'une expertise centralisée.
Architecture de base	Basée principalement sur des produits Open source : <ul style="list-style-type: none"> - Linux - Apache - Tomcat - etc.
Etat de la communauté	<ul style="list-style-type: none"> - Vitalité de la communauté. - Qualité des échanges. - Stabilité des leaders. - Structure d'affaires.
Support commercial	Présence d'entreprises basées au Québec qui peuvent offrir un support en français à tous les niveaux.
Support multilingue	<ul style="list-style-type: none"> - Interface usager minimalement en trois langues: français, anglais, espagnol. - Facilité de traduction (séparation des messages et code). - Pas de limite de langue pour publication des pages HTML. - Support de date en format international : configurable. - En tout temps, l'utilisateur doit être capable de choisir la langue. - Fortement souhaité: l'enseignant peut imposer une langue
Standards	Bonne utilisation des standards et une volonté ferme de continuer à évoluer dans ce sens.
Support et documentation	<ul style="list-style-type: none"> - Documentation contextuelle et de qualité pour enseignants et étudiants. - Documentation technique de qualité. - Support rapide.
Base de données	Couche abstraite - Possibilités MySQL - Oracle
Modularité et adaptation	Les modules sont faiblement liés les uns aux autres et il existe une cohérence d'un module à l'autre. Les possibilités d'adaptation sont bien documentées (programmation).
Gestion de la croissance	Le système peut être déployé sur plusieurs serveurs avec un mécanisme pour répartir la charge. Le système peut être utilisé plusieurs serveurs de base de données. Existence de quotas par cours.
Sécurité	<ul style="list-style-type: none"> - Possibilité de préserver la confidentialité des données (https) - Implantation de mécanismes pour tracer les activités et contrôle en temps réel. - Accès aux fichiers par le serveur Web uniquement - Mécanisme pour authentifier (cookies et sessions). - Infrastructure d'autorisation : rôles - Validation de l'input - Anti-virus
Robustesse et fiabilité	<ul style="list-style-type: none"> - Pas trop dépendant de JavaScript (et si oui ECMAScript 262) - Copies de sauvegarde par cours (enseignant, administrateur) - Pas d'arrêt du système lors des prises de copies de sécurité, ou lors de ré-indexation . - Restauration de cours facile
Accès par fureteur	Clients Web - Standard Web multi-fureteur

3.3.2. Importation d'un paquet de contenus en norme de SCORM 2004 sur la plate-forme Moodle

Moodle est un logiciel libre, écrit en langage PHP avec la base de données de MySQL. Elle est utilisée par **plus de 3000 organismes dans 160 pays** et le nombre d'utilisateurs est de plus en plus augmentable de 10% par mois. On peut télécharger Moodle dans le site <http://moodle.org>.

Pour importer un cours en norme de SCORM 2004 sur cette plate-forme là on peut suivre des étapes ci-dessous :

- 1) Entrer Moodle sous Administrateur.
- 2) Créer un nouveau cours en utilisant la fonction :  Courses
- 3) Choisir une catégorie de cours => Add a new course.
- 4) Entrer le cours créé => Teacher => Choisir ce professeur pour le cours.
- 5) Entrer sous le nom du professeur => Démarrer la fonction Edition.
- 6) On choisit «Add an activity » et ensuite on choisit SCORM/AICC.
- 7) Remplacer toutes les informations nécessaires et choisit « Choose or update a package » pour parcourir le cours en paquet de SCORM 2004 qui vient d'être créé => Upload a file.
- 8) Marquer le paquet choisi et choisit « Choose » => Save change.

Chapitre 4

Transformation automatique d'un cours local à la norme de SCORM 2004 en utilisant l'éditeur Reload

4.1. Étude d'éditeur Reload

Reload Editor [29] est un outil libre écrit en Java permettant d'emballer le contenu et d'éditer les méta-données, les règles de séquençement d'un paquet SCORM. Avec Reload Editor, nous pouvons créer, importer, rédiger, et exporter les paquets de contenu en norme SCORM.

Reload Editor ressemble à la plupart des autres progiciels. Une sous fenêtre permet de rédiger, de construire un paquet SCORM (voir figure 4.1).

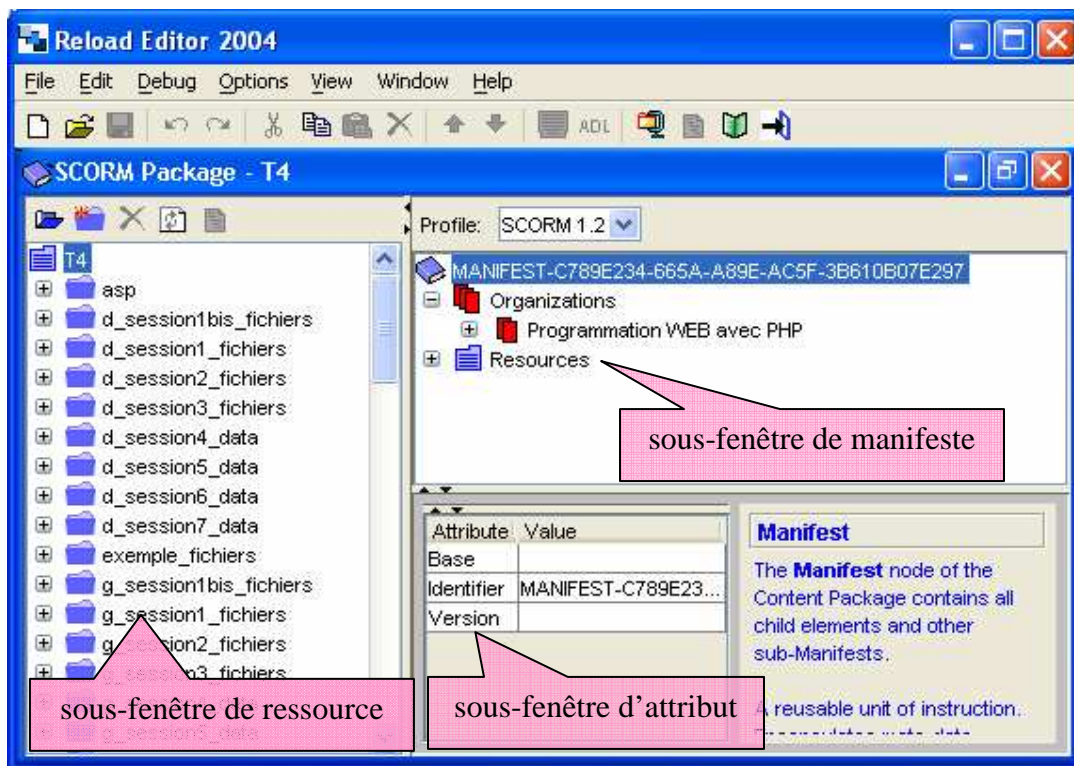


Figure 4.1 : Fenêtre principale du Reload Editeur

Cet outil a les trois fonctions essentielles (voir figure 4.2):

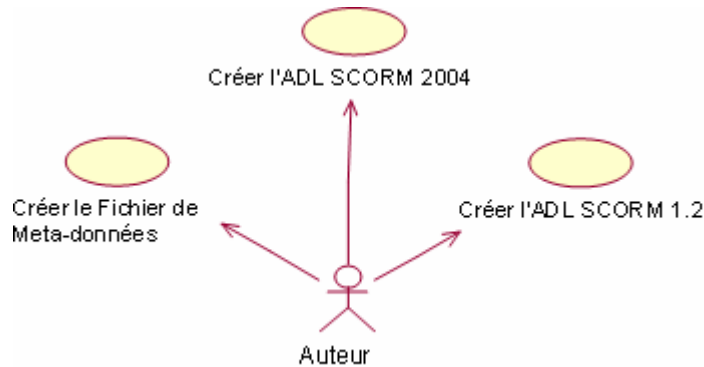


Figure 4.2 : Les fonctions du Reload Editeur

Ici on s'intéresse à la fonction Créer l'ADL SCORM 2004. Elle a des sous-fonctions comme suivantes :

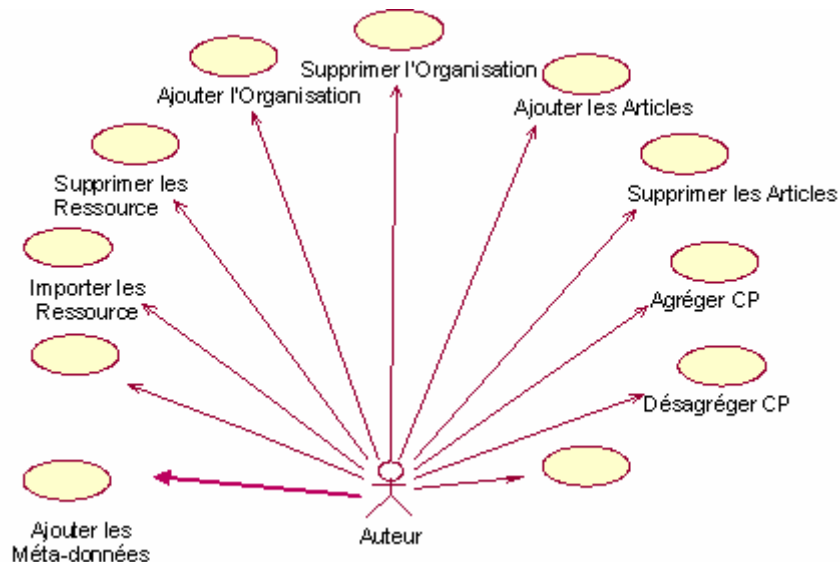


Figure 4.3 : Les sous fonctions de la fonction Créer l'ADL SCORM 2004

Problème 5 : Après d'analyser les fonctions du Reload Editeur on trouve qu'une fonction de transformer des fichiers de contenu en SCO **lui manque**. Est-ce que on peut ajouter une fonction de transformer les fichiers de contenu en SCO à la norme de SCORM 2004 et Comment on peut l'intégrer dans Reload Editeur (voir section 4.2)?

4.2. Etude technique du programme à modifier

4.2.1. Description

Pour pouvoir transformer un cours à la norme SCORM on doit ajouter des fonctions permettant de parcourir tous les fichiers à transformer pour structurer ce cours là selon l'idée de professeur, ainsi que d'ajouter automatique tous les fonctions permettant aux SCO de communiquer avec les LMS (en temps d'exécute) et les règles sur lesquelles les activités d'apprentissage basent pour compléter ce cours.

4.2.2. Scénario d'utilisation du programme

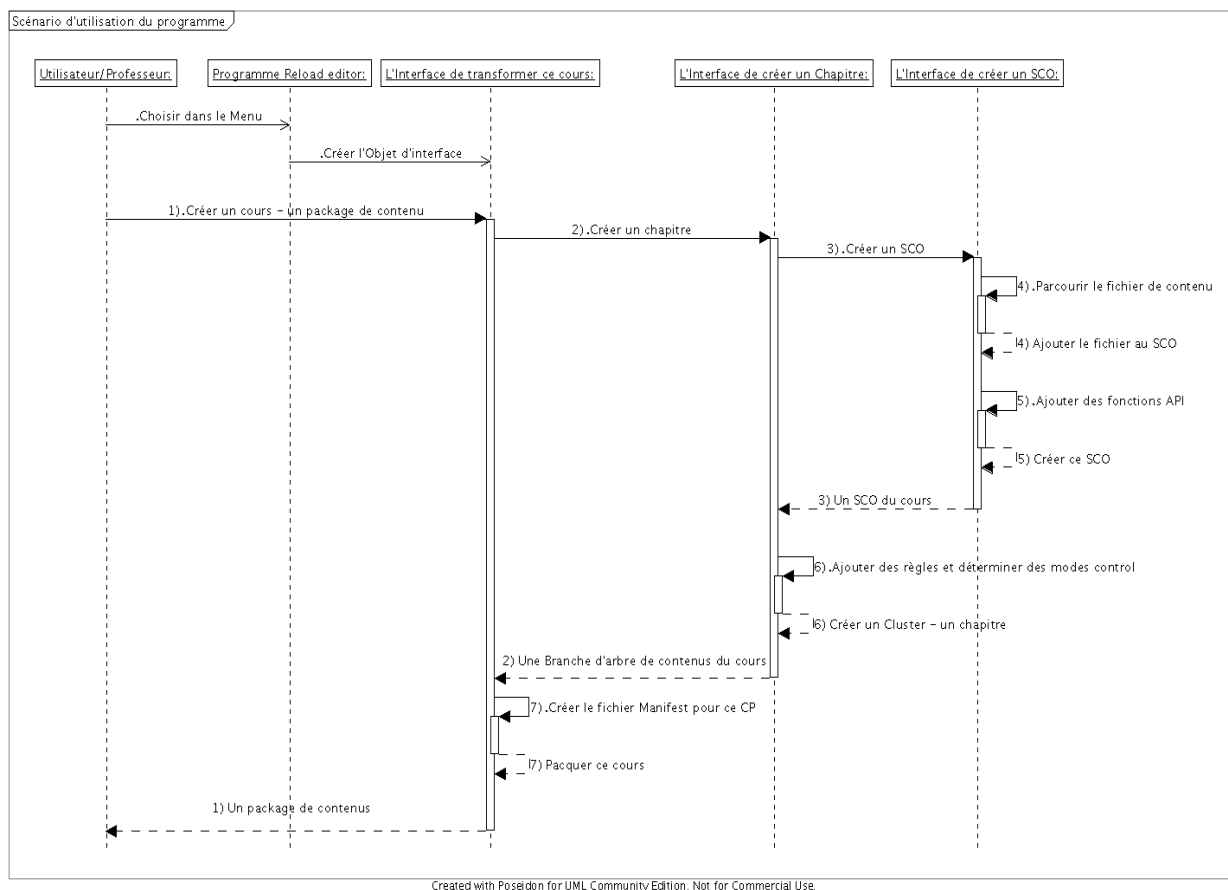


Figure 4.4 : Scénario d'utilisation du programme

4.2.3. Insertion automatique des fonctions aux SCO

Dans cette section on résout le problème 3 qu'on a posé en chapitre 3 : Comment on peut insérer **automatique** des fonctions JavaScripts aux SCO ?

Dans chaque SCO, il y faut avoir au moins deux fonctions : Initialize() et Terminate() pour commencer et terminer la communication avec LMS. Dans ce travail on a construit les deux librairie en deux couches (voir chapitre 3 et l'Annexe A): Couche de système et couche d'application. L'insertion de ces fonctions dans chaque SCO est, donc, des appels d'eux.

Problème 6 : Comment on peut traiter des fonctions JavaScripts qui existent dans ces fichiers avant ?

Solution : Chaque fois on ajoute un fichier de contenu au SCO on doit trouver tous les liens et fonctions dans ce fichier et les copie dans le répertoire du SCO.

Problème 7 : Où on peut insérer des appels dans les fichiers *.html ?

Solution : Pour que l'insertion des fonctions ajoutées n'influe pas sur les fonctions qui existent on peut ajouter les appels au début ou à la fin du fichier. Dans ce travail nous choisissons la deuxième solution.

Problème 8 : Chaque SCO n'est initié qu'une fois au début de la communication avec LMS. Dans le cas : un SCO contient plusieurs pages. Comment on peut le traiter ?

Solution : On utilise l'ordre des fichiers qui est trié par le professeur. On ajoute l'appel à la fonction Initialize() seulement pour la première page et à la fonction Terminate() seulement pour la dernière page.

Problème 9 : Avec des SCO contenant plusieurs pages comment on peut parcourir ce SCO ?

Solution : On ajoute des boutons et des fonctions JavaScripts dans chaque fichier qui permettent aux utilisateurs de parcourir ce SCO.

Problème 10 : Dans le moment où l'apprenant fait un Essai sur un SCO, il peut noter ce qu'il lit dans chaque page. Comment on peut le faire ?

Solution : On crée les deux pages « template » pour noter et afficher les idées propres des apprenants. Dans ces deux pages on ajoute des fonctions nécessaires pour communiquer avec LMS via des éléments de cmi.comments_from_learner (voir chapitre 3).

Problème 11 : Chaque variable en JavaScripts est effective dans cette page où elle « habite ». Comment on peut retourner la page précédente après avoir noté ?

Solution : On utilise les « Cookies » pour marquer chaque page.

Problème 12 : Comment on peut déterminer la complétion du SCO ?

Solution : Ça dépend de chaque type de SCO. Dans ce cas la complétion du SCO est la lecture de la dernière page et on utilise les éléments de `cmi.exit`, `cmi.entry` et `cmi.completion_status` pour communiquer avec LMS (voir chapitre 3).

Problème 13 : Dans le moment où l'apprenant fait un Essai sur un SCO, il peut suspendre ou quitter ce SCO. Comment on peut le gérer ?

Solution : On ajoute des boutons dans chaque page qui permettent aux apprenants de suspendre ou quitter ce SCO en utilisant des éléments de `cmi.exit` et `cmi.entry` pour communiquer avec LMS. Dans ce cas de « resume » on peut utiliser l'élément de `cmi.location` pour déterminer la page où on a suspendu à la fois précédente.

4.2.4. Fenêtre principale du programme modifié

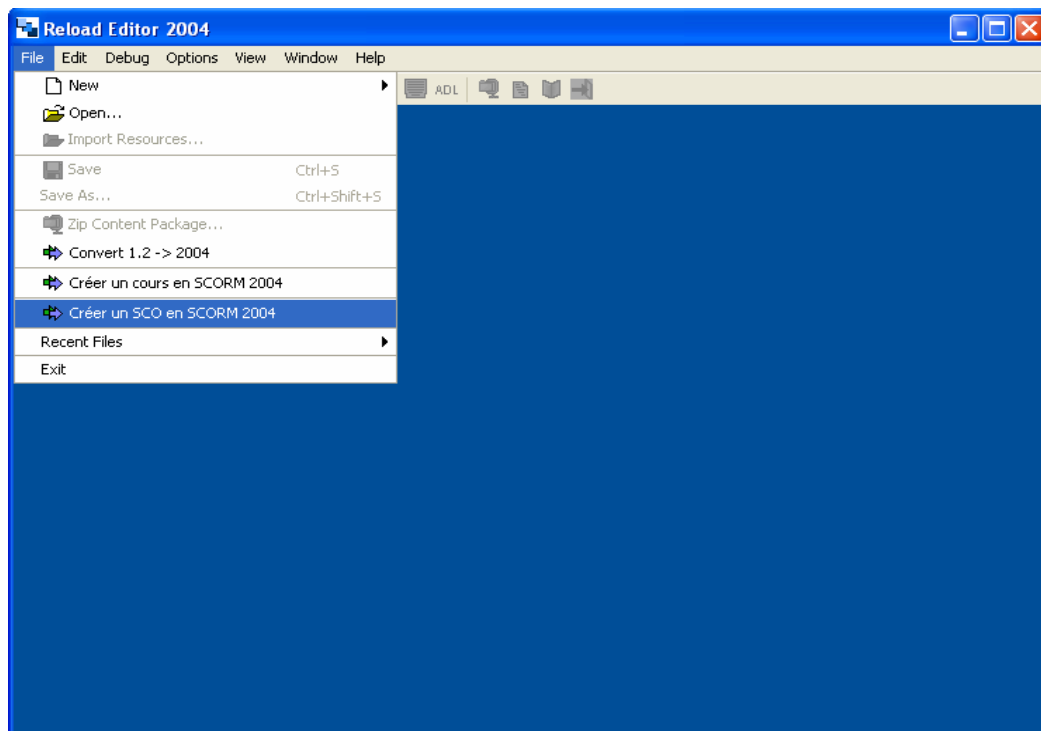


Figure 4.5 : Les deux fonctions ajoutées dans l'Editeur Reload

4.3. Processus de la transformation automatique d'un cours local à la norme de SCORM 2004 en utilisant l'éditeur Reload modifié

4.3.1. Transformation des fichiers du cours aux SCO

On utilise la fonction ajoutée : File->Créer un SCO en SCORM 2004.
On a une fenêtre ci-dessous :

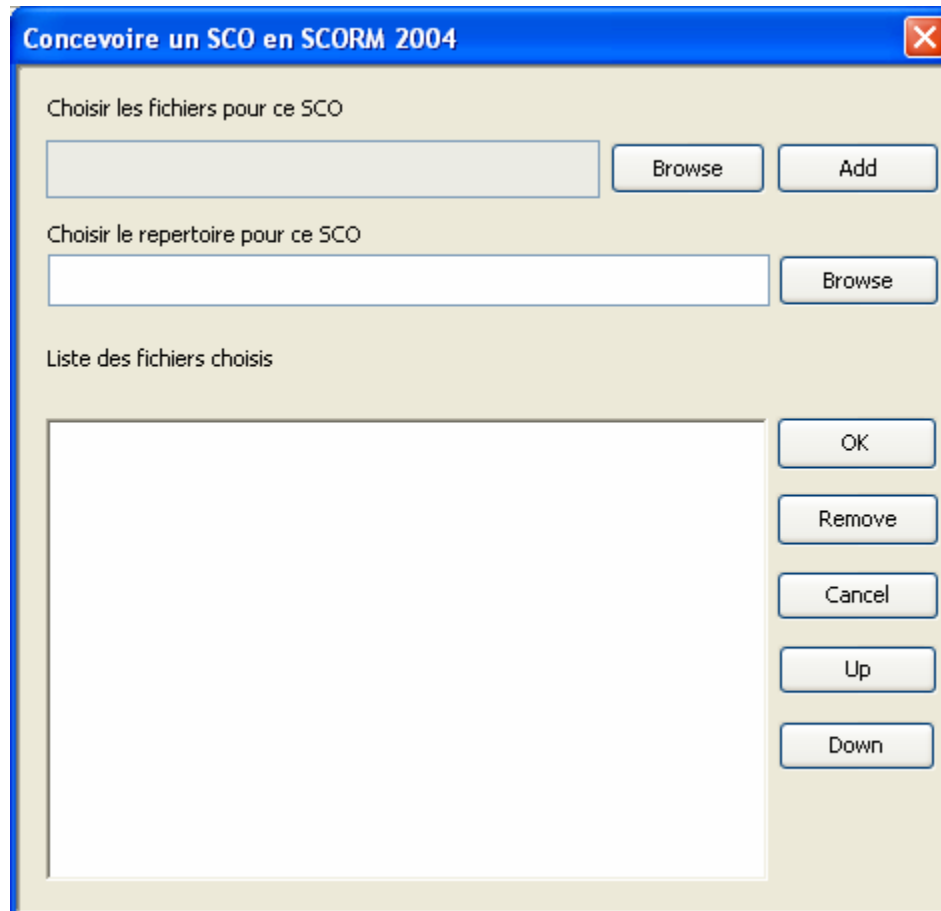


Figure 4.6 : Construction d'un SCO à partir des fichiers du cours

Avec la fonction on peut faire des activités nécessaires pour construire un SCO : Déterminer le répertoire où on sauvegarde ce SCO ; chercher des fichiers du cours ; ajouter ou enlever des fichiers dans ce SCO ; modifier l'ordre des fichiers dans ce SCO (l'ordre des fichiers est l'ordre des pages dans ce SCO)...

Chaque fois on ajoute un fichier dans ce SCO tous les liens et les fonctions existant dans ce fichier sont ajoutés dans le répertoire du SCO.

Quand on presse sur le bouton « OK » tous les fichiers dans la liste est ajouté avec des fonctions nécessaires pour pouvoir communiquer avec LMS dans l'ordre précisé par le professeur et on a un SCO créé. On peut refaire cette étape plusieurs fois pour construire les autres SCO.

4.3.2. Construction d'un Paquet de SCORM 2004

On utilise la fonction ajoutée : File->Créer un cours en SCORM 2004 (cette fonction réutilise des classe existant du Reload). Puis choisit un répertoire pour le cours. On a la fenêtre ci-dessous :

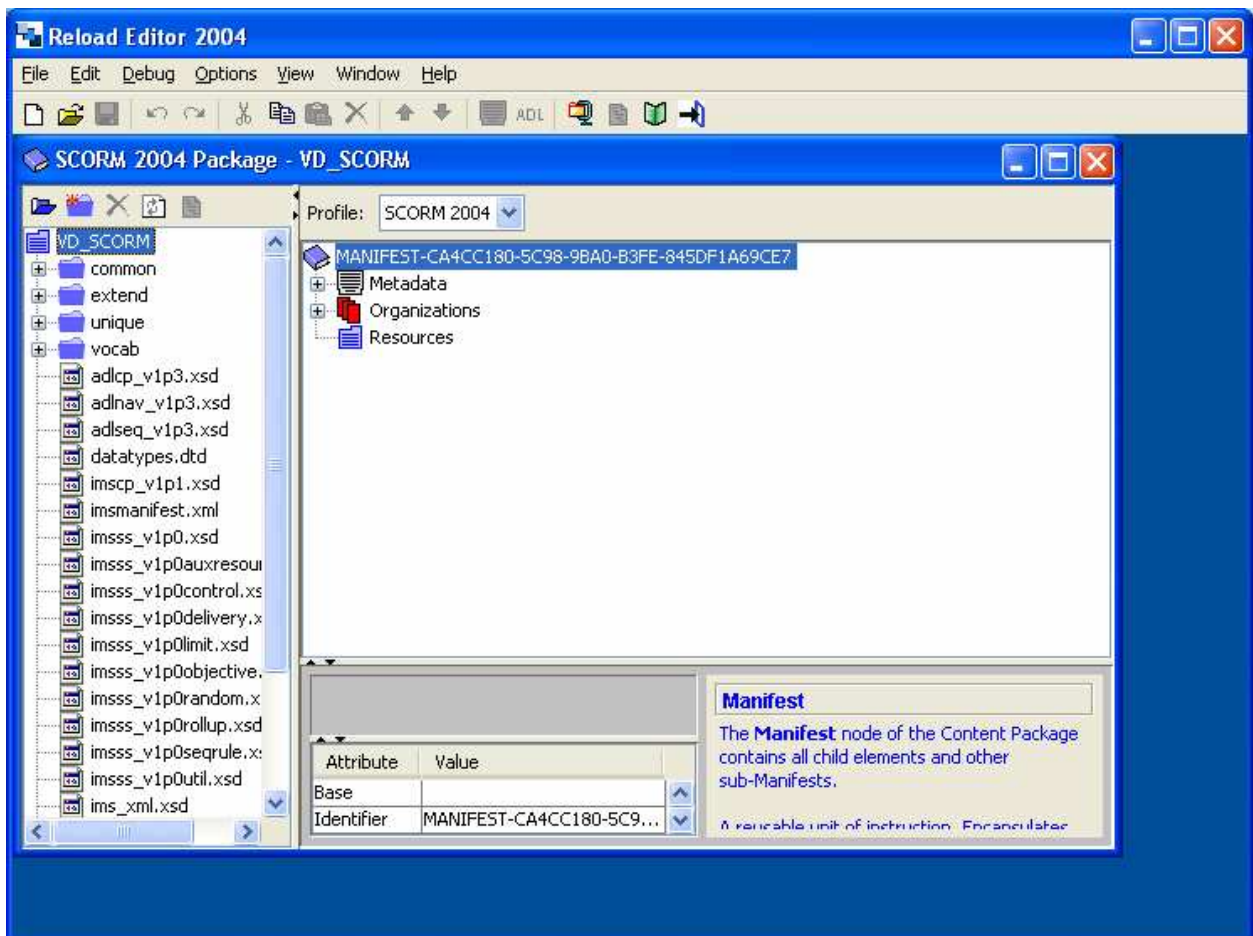


Figure 4.7 : La fenêtre de construction d'un paquet de SCORM 2004

Dans la sous fenêtre à gauche on peut créer des sous répertoires pour le cours et ou pour les chapitres (en utilisant la fonction « New folder »). Dans chaque sous répertoire on peut choisir des SCO en utilisant la fonction « Import Resources » (voir figure 4.8). Refaites pour tous les SCO nécessaires dans ce cours.

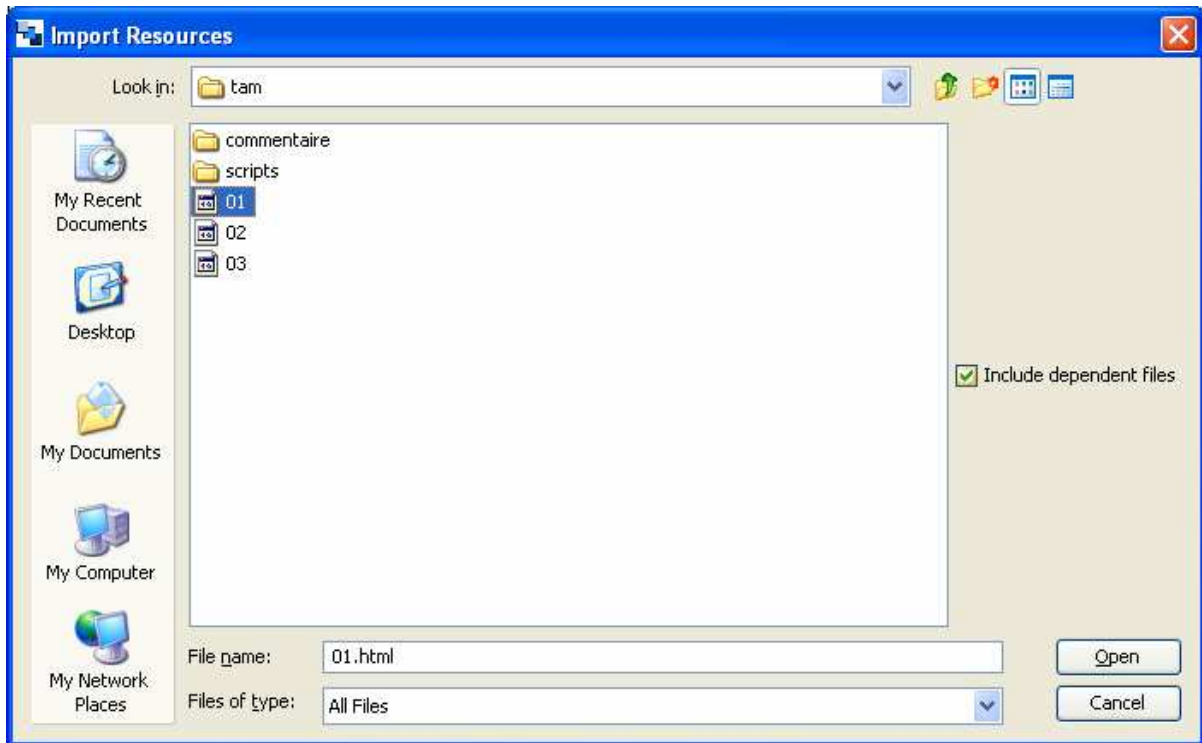


Figure 4.8 : Importation des SCO dans le cours

Fait attention : Pour chaque SCO on choisit la première page à ajouter dans le cours et vérifiez que l’option « Include dependent files » est choisie pour ajouter tous les fichiers, répertoires et des bibliothèques nécessaires au cours.

Après avoir ajouté tous les SCO dans le cours, on utilise la fonction « Zip Content Package... » pour construire un paquet de contenu en norme de SCORM 2004 qui peut être importé sur une plate-forme.

4.3.3. Importation d’un Paquet de SCORM 2004 sur Moodle

On le fait comme la section 3.3.2 dans le chapitre 3 avec le paquet créé.

Conclusion

La Formation Ouverte et à Distance (FOAD) existe depuis longtemps (l'année même de l'apparition du timbre-poste) basé sur la autoformation des apprenants et elle porte le cachet des épisodes historiques de développement de la technologie. Elle est réalisée en utilisant tous les moyens possibles comme : la poste, le téléphone, la radio, la télévision...

Dans ce travail nous avons fait des études concernant la FOAD dans l'ère de l'Internet. En effet, nous avons étudié en détail le modèle SCORM 2004, qui est un ensemble de spécifications basées sur plusieurs normes et standards connus comme : Les spécifications des méta-données de l'IEEE, le modèle de données et l'API de l'IEEE, la structure du contenu de l'AICC et le conditionnement du contenu de l'IMS **pour résoudre le problème de partage des ressources** entre les LMS conformant les modèles différents.

Dans le cadre de ces TRAVAUX D'INTÉRÊT PERSONNEL ENCADRÉ nous avons divisé ce travail en les deux parties :

- **Partie théorique** : Dans cette partie nous avons compris toutes les spécifications du modèle SCORM 2004 qui sont décrites dans environs 2000 pages de document et appris comment fonctionne ce modèle là.
- **Partie pratique** : Dans cette partie nous avons fait des études et trouvé les processus principaux de transformation d'un cours en ligne à la norme de SCORM 2004. Nous avons réussi à modifier l'éditeur Reload, qui permet aux professeurs d'emballer des Assets ou des SCO existant dans un paquet de SCORM 2004, qui peut être importé à une plate-forme conformant le modèle SCORM comme Moodle. En effet, cet éditeur ne peut pas créer des nouveaux SCO à partir des fichiers du cours. Dans ce travail nous

avons essayé et réussi à ajouter une fonction dans Reload permettant de pouvoir créer des nouveaux SCO automatiquement.

Toutefois, il y a des travaux à améliorer ce programme à l'avenir pour:

- voir directement les résultats de transformer dans les fenêtres.
- transformer n'importe quel fichier qu'on veut.
- transformer un cours en ligne directement.
- créer un nouveau cours en norme de SCORM 2004

BIBLIOGRAPHIE

- [1] Blandin, Bernard. Historique de la formation ouverte et à distance. Actualité de la formation permanente, n° 189, mars-avril 2004, p. 6 9-71
- [2] Gord, Mackenzie. SCORM 2004 Primer. A (Mostly) Painless Introduction to SCORM. Version 1.0. Senior Business Strategist, August 03, 2004 (25 p.)
www.mcgill.com/media/SCORM_2004_Primer_v1_McGill_Digital_Solutions_Gord_Mackenzie.pdf
- [3] EDucNET. Glossaire e-Formation. Mis à jour mai 2003.
www.educnet.education.fr/superieur/glossaire.htm
- [4] Isabelle GALY - Patrice MAGNARD. Le e-learning ou e-formation. Rapport du Club.Senat.fr, 26 février 2003
- [5] FUTURA Sciences. Concept, Design & Contenu © 2001-2006 Futura-Sciences.com
http://www.futura-sciences.com/comprendre/g/definition-norme_1263.php
http://www.futura-sciences.com/comprendre/g/definition-standard_1325.php
- [6] Le e-learning adaptatif. Université de Nice Sophia-Antipolis. Master recherche PMLT. Rapport de stage. Présenté en juin 2005 par Sterenn AUBERT
- [7] SCTIC : Les normes et standards de la formation en ligne. État des lieux et enjeux, 2002
- [8] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Le modèle de référence SCORM^{MC} (*Sharable Content Object Reference Model*), Version 1.2 Aperçu de SCORM, 1er octobre 2001
Disponible sur ADLNet.org (<http://www.adlnet.org>)
- [9] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM®) 2004 2nd Edition Overview, July 22 2004
Available at ADLNet.org (<http://www.adlnet.org>)
- [10] Kirk Johnson, ADL Technical Center Steve Slosser, Joint ADL Co-Lab. SCORM Status and Evolution. International Plugfest II January 17, 2006
- [11] Fujitsu learning media limited (FML). The program for manager of e-Learning Centres, 2005
- [12] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM®) 2004. Content Aggregation Model (CAM) Version 1.3.1, July 22 2004
Available at ADLNet.org (<http://www.adlnet.org>)

- [13] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM®) 2004. Content Aggregation Model (CAM) Version 1.3.2, February 13 2006
Available at ADLNet.org (<http://www.adlnet.org>)
- [14] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM®) 2004. Run-Time Environment (RTE) Version 1.3.1, July 22 2004
Available at ADLNet.org (<http://www.adlnet.org>)
- [15] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM®) 2004. Run-Time Environment (RTE) Version 1.3.2, February 13 2006
Available at ADLNet.org (<http://www.adlnet.org>)
- [16] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM®) 2004. Sequencing and Navigation (SN) Version 1.3.1, July 22 2004
Available at ADLNet.org (<http://www.adlnet.org>)
- [17] Advanced Distributed Learning Initiative. Projet Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM®) 2004. Sequencing and Navigation (SN) Version 1.3.2, February 13 2006
Available at ADLNet.org (<http://www.adlnet.org>)
- [18] Blandin, Bernard. Formations Ouvertes et A Distance : l'opportunité de ré-interroger nos pratiques, 18 novembre 2003
- [19] NGO Anh Tuan. On-line interaction in collaborative and problem-based learning approach. The 2nd ICT in Education, international Conference/ Expo in Vietnam, 2004
- [20] Prof. Dr. Ullrich Dittler. E-Learning, Einsatzkonzepte und Erfolgsfaktoren des Lernens mit interaktiven Medien. 2. überarbeitete und ergänzte Auflage, Oldenbourg Verlag München Wien, 2003
- [21] Wolfgang Ihbe. Lernen mit Multimedia, ein Beitrag zur Förderung beruflicher Handlungskompetenzen, 2003
- [22] TS. Nguyễn Quang Trung. E-Learning: Phương thức đào tạo cho tương lai và giải pháp sử dụng phần mềm nguồn mở. Học viện Công nghệ Bưu chính Viễn thông, 2005.
- [23] Morgan Keegan. ELearning, the engine of the knowledge economy. Members New York Stock Exchange, 2004
- [24] Fred Lockwood. Open and distance learning today. Routledge studies in distance education, London and New York, 1995
- [25] Richard Wyles. NZ Open Source Virtual Learning Environment, New Zealand, 2005
- [26] Université du Québec à Montréal. Rapport du comité institutionnel sur les plateformes d'apprentissage en ligne, 2005

- [27] Desmond Keegan. Foundation of distance education. Third edition. Routledge, London and New York, 1996
- [28] CMI Guidelines for Interoperability AICC. DOCUMENT NO. CMI001. Revision 4.0 release 16-Aug-2004
- [29]. Reload Editor, Tool for content packaging, available at: <http://www.reload.ac.uk>

Des URL intéressants sur Internet:

- [1] <http://www.educnet.education.fr/superieur/normes.htm>
- [2] <http://ieeeltsc.org/>
- [3] <http://www.imsproject.org/>
- [4] <http://www.ariadne-eu.org/>
- [5] <http://www.aicc.org/>
- [6] <http://www.adlnet.org/>
- [7] <http://tice.education.fr/educnet/>
- [8] <http://www.macromedia.com/resources/elearning/standards.html>
- [9] *Reload Editor 2004B*, <http://www.sourceforge.net/projects/x4l-reload>
- [10] Les définitions dans <http://www.futura-sciences.com/>
- [11] <http://moodle.org>

ANNEXE A Les deux bibliothèques du programme

Les bibliothèques en deux couches sont sauvegardées dans le répertoire /scripts comme suivant :

```
/******  
** La couche de système – Le fichier APIWrapper.js  
*****/  
  
// locale variable pour chercher l'objet d'API  
var apiHandle = null;  
var findAPITries = 0;  
var noAPIFound = "false";  
  
var terminated = "false";  
var _debug = false;  
  
/******  
**  
** La fonction cherche l'objet d'API pour le SCO  
**  
** Retourné: L'Object d'API ou null si on ne le trouve pas.  
**  
*****/  
function findAPI( win )  
{  
  while ( (win.API_1484_11 == null) &&  
    (win.parent != null) &&  
    (win.parent != win) )  
  {  
    findAPITries++;  
    if ( findAPITries > 500 )  
    {  
      alert( "Error finding API -- too deeply nested." );  
      return null;  
    }  
    win = win.parent;  
  }  
  return win.API_1484_11;  
}  
  
/******  
**  
** Entrée: Rien  
**  
** Retourné: L'Object d'API ou null si on ne le trouve pas
```

```
**
*****/
function getAPI()
{
    var theAPI = findAPI( window );

    if ( (theAPI == null) &&
        (window.opener != null) &&
        (typeof(window.opener) != "undefined") )
    {
        theAPI = findAPI( window.opener );
    }

    if (theAPI == null)
    {
        alert( "Unable to locate the LMS's API Implementation.\n" +
            "Communication with the LMS will not occur." );
        noAPIFound = "true";
    }

    return theAPI
}

/*****
**
** Entrée: Rien
**
** Retourné: Objet - apiHandle variable.
**
*****/
function getAPIHandle()
{
    if ( apiHandle == null )
    {
        if ( noAPIFound == "false" )
        {
            apiHandle = getAPI();
        }
    }
    return apiHandle;
}

/*****
**
** La fonction utilise API.Initialize() pour amorcer la communication
**
** Entrée: Rien
```

```
**
** Retourné: String - "true" si succès, ou "false" au contrat.
**
*****/
function initializeCommunication()
{
    var api = getAPIHandle();

    if ( api == null )
    {
        return "false";
    }
    else
    {
        var result = api.Initialize("");

        if ( result != "true" )
        {
            var errCode = retrieveLastErrorCode();

            displayErrorInfo( errCode );

            // may want to do some error handling
        }
    }

    return result;
}

/*****
**
** La fonction utilise API.Terminate() pour terminer la communication
**
** Entrée: Rien
**
** Retourné: String - "true" si succès ou "false" si au contrat.
**
*****/
function terminateCommunication()
{
    var api = getAPIHandle();

    if ( api == null )
    {
        return "false";
    }
    else
```

```
{
// call Terminate only if it was not previously called
if ( terminated != "true" )
{
// call the Terminate function that should be implemented by
// the API
var result = api.Terminate("");

if ( result != "true" )
{
var errCode = retrieveLastErrorCode();

displayErrorInfo( errCode );

// may want to do some error handling
}
else // terminate was successful
{
terminated = "true";
}
}
}

return result;
}

/*****
**
** Entrée: String - Nom d'un élément du modèle de données
**          (e.g. cmi.core.learner_id)
**
** Retourné: String - La valeur correspondante au LMS
**
*****/
function retrieveDataValue( name )
{
var api = getAPIHandle();
if ( api == null )
{
return "";
}
else
{
var value = api.GetValue( name );
var errCode = api.GetLastError();
if ( errCode != "0" )
{
```

```
        var errCode = retrieveLastErrorCode();

        displayErrorInfo( errCode );
    }
    else
    {
        return value;
    }
}
return;
}

/*****
**
** Entrée: String - Nom d'un élément du modèle de données
**
**      String - La valeur correspondante à l'élément
**
** Retourné: String - "true" si succès ou "false" si au contrat
**
*****/
function storeDataValue( name, value )
{
    // do not call a set after finish was called
    if ( terminated != "true" )
    {
        var api = getAPIHandle();

        if ( api == null )
        {
            return;
        }
        else
        {
            var result = api.SetValue( name, value );

            if ( result != "true" )
            {
                var errCode = retrieveLastErrorCode();

                displayErrorInfo( errCode );

                // may want to do some error handling
            }
        }
    }
}
```

```
    return;
}

/*****
**
** Entrée: Rien
**
** Retourné: String - Le code d'erreur.
**
*****/
function retrieveLastErrorCode()
{
    // It is permitted to call GetLastError() after Terminate()

    var api = getAPIHandle();

    if ( api == null )
    {
        return "";
    }
    else
    {
        return api.GetLastError();
    }
}

/*****
**
** Entrée: String - Le code d'erreur.
**
** Retourné: String - La description de l'erreur
**
*****/
function retrieveErrorInfo( errCode )
{
    // It is permitted to call GetLastError() after Terminate()

    var api = getAPIHandle();

    if ( api == null )
    {
        return "";
    }
    else
    {
```

```
        return api.GetErrorString( errorCode );
    }
}

/*****
**
** Entrée: String - Le code d'erreur.
**
** Retourné: String - Les informations devinant sur le code d'erreur.
**
*****/
function retrieveDiagnosticInfo( error )
{
    // It is permitted to call GetLastError() after Terminate()

    var api = getAPIHandle();

    if ( api == null )
    {
        return "";
    }
    else
    {
        return api.GetDiagnostic( error );
    }
}

/*****
**
** La fonction demande au LMS de sauvegarder toutes les informations dans le tampon
**
*****/
function persistData()
{
    // do not call a set after Terminate() was called
    if ( terminated != "true" )
    {
        var api = getAPIHandle();

        if ( api == null )
        {
            return "";
        }
        else
        {
            return api.Commit();
        }
    }
}
```

```
}
else
{
    return "";
}
}

/*****
**
** La fonction pour afficher l'erreur à l'écran
**
** Entrée: String - Le code d'erreur
**
*****/
function displayErrorInfo( errCode )
{
    if ( _debug )
    {
        var errString = retrieveErrorInfo( errCode );
        var errDiagnostic = retrieveDiagnosticInfo( errCode );

        alert( "ERROR: " + errCode + " - " + errString + "\n" +
            "DIAGNOSTIC: " + errDiagnostic );
    }
}

/*****
** La couche d'application – Le fichier SCOFonctions.js
*****/
var IntraNavigation = false
var NomDePage = ""

// La fonction pour marquer la page en Cookie
function setNomDePage( value )
{
    SetCookie("NomDePage", value, 1);
}

// La fonction pour lire la marque de la page dans la Cookie
function getNomDePage()
{
    NomDePage = ReadCookie("NomDePage");
}

/*****
** La fonction pour déplacer dans SCO ayant plusieurs pages
```

```
*****/
function GoToPage( page )
{
    IntraNavigation = true;

    // replace the current page with the page specified
    window.location.replace( page );
}

/*****
*****/
function renderPreviousButton()
{
    var value = retrieveDataValue( "adl.nav.request_valid.previous" );
    return value;
}

/*****
*****/
function renderContinueButton()
{
    var value = retrieveDataValue( "adl.nav.request_valid.continue" );
    return value;
}

/*****
*****/
function Previous()
{
    // we request the previous SCO from the LMS
    storeDataValue("adl.nav.request", "previous");

    // we terminate this SCO's communication with the LMS
    terminateCommunication();
}

/*****
*****/
function Continue()
{
    // we request the next SCO from the LMS
    storeDataValue ("adl.nav.request", "continue");

    // we terminate this SCO's communication with the LMS
    terminateCommunication();
}

```

```

/*****
*****/
function onUnexpectedExit( value )
{
    // set a bookmark
    storeDataValue( "cmi.location", value );
}

/*****
*****/
function displayNext()
{
    document.getElementById("nextBtn").style.visibility = "visible";
}

/*****
*****/
function Initialize(showNext)
{
    if ( !(entryStatus == "resume" )
    {
        initializeCommunication();
    }

    // if there is somewhere to go next and previous enable the buttons
    if ( renderContinueButton() == "true" )
    {
        if ( showNext != "false" )
            document.getElementById("nextBtn").style.visibility = "visible";
    }

    if ( renderPreviousButton() == "true" )
    {
        document.getElementById("prevBtn").style.visibility = "visible";
    }

    // we need to determine if this is a new "learner attempt" or a
    // suspended "learner attempt
    var entryStatus = retrieveDataValue( "cmi.entry" );

    if ( entryStatus == "resume" )
    {
        var location = retrieveDataValue( "cmi.location" );
        // jump to the location we just retrieved

```

```
        window.location.replace( location );
    }
}

/*****
*****/
function Terminate()
{
    disablePrevious();
    disableNext();
    disableNote();
    disableNoteDisplay();
    if (!IntraNavigation)
    {
        storeDataValue( "cmi.completion_status", "completed" );

        storeDataValue( "cmi.exit", "" );

        terminateCommunication();
    }
}

function disableSubmit()
{
    document.getElementById('submit').disabled = true;
    document.getElementById('reset').disabled = true;
}

function disablePrevious()
{
    document.getElementById('prevBtn').disabled = true;
}

function disableNext()
{
    document.getElementById('nextBtn').disabled = true;
}

function disableNote()
{
    document.getElementById('noteBtn').disabled = true;
}

function disableNoteDisplay()
{

```

```
        document.getElementById('displayBtn').disabled = true;
    }

function enableSubmit()
{
    document.getElementById( "submit" ).disabled = false;
    document.getElementById( "reset" ).disabled = false;
}

function enablePrevious()
{
    document.getElementById( "prevBtn" ).disabled = false;
}

function enablePrevious()
{
    document.getElementById( "nextBtn" ).disabled = false;
}

function getDateTime()
{
    var now = new Date;
    var yyyy = now.getFullYear();
    var mm = Number( now.getMonth() ) + 1;
    if ( mm < 10 ) mm = "0" + mm;
    var dd = now.getDate();
    if ( dd < 10 ) dd = "0" + dd;
    var hh = now.getHours();
    if ( hh < 10 ) hh = "0" + hh;
    var mn = now.getMinutes();
    if ( mn < 10 ) mn = "0" + mn;
    var ss = now.getSeconds();
    if ( ss < 10 ) ss = "0" + ss;
    var timestamp = yyyy+"-"+mm+"-"+dd+"T"+hh+": "+mn+": "+ss;
    //var timestamp = dd+"-"+mm+"-"+yyyy+"Time : "+hh+": "+mn+": "+ss;
    return timestamp;
}

/*
// Set et read cookie
*/

function SetCookie(cookieName,cookieValue,nDays) {
    var today = new Date();
    var expire = new Date();
    if (nDays==null || nDays==0) nDays=1;
    expire.setTime(today.getTime() + 3600000*24*nDays);
```

```
document.cookie = cookieName+"="+escape(cookieValue)
    + ";expires="+expire.toGMTString();
}

function ReadCookie(cookieName) {
    var theCookie=""+document.cookie;
    var ind=theCookie.indexOf(cookieName);
    if (ind==-1 || cookieName=="") return "";
    var ind1=theCookie.indexOf('; ',ind);
    if (ind1==-1) ind1=theCookie.length;
    return unescape(theCookie.substring(ind+cookieName.length+1,ind1));
}

var status;

function Exit ( value )
{
    //alert (status);
    if ((status == -1) || (status == -2)) // il y a 1 page ou derniere page
    {
        Terminate();
        GoToPage("");
    }
    else
    {
        storeDataValue( "cmi.completion_status", "incomplete" );
        storeDataValue( "cmi.exit", "" );
        onUnexpectedExit( value );
        terminateCommunication();
        GoToPage("");
    }
}

function Suspend ( value )
{
    storeDataValue( "cmi.completion_status", "incomplete" );
    storeDataValue( "cmi.exit", "suspend" );
    onUnexpectedExit( value );
    terminateCommunication();
    GoToPage("");
}

/*****
*****/
function AjouterLesFonctions( value )
```



```
{
    status = value;
    if (value == 0) // Premiere page
    {
        Initialize();
    }
    if (value == -1) // 1 page seulement
    {
        Initialize();
    }
    if (value == -2) // Derniere Page
    {
    }
}
```

ANNEXE B Les éléments du modèle de données de SCORM RTE

Version	
cmi._version	<p>Description : représente la version du modèle de données.</p> <p>Par exemple :</p> <p>GetValue("cmi._version")</p>
Commentaire de l'apprenant	
cmi.comments_from_learner._children	<p>Description : Cet élément représente une liste des éléments soutenus de modèle de données. Cet élément de modèle de données est typiquement employé par un SCO pour déterminer quels éléments de modèle de données sont soutenus par le LMS. Une chaîne de caractères retournée peut être employé par le SCO pour établir dynamiquement des paramètres pour les demandes de GetValue() et de SetValue().</p> <p>Par exemple :</p> <p>GetValue("cmi.comments_from_learner._children")</p>
cmi.comments_from_learner._count	<p>Description : Cet élément décrit le nombre courant de commentaires d'étudiant qui sont stockés par le LMS pour le SCO. Le nombre au total sera retourné</p> <p>Par exemple :</p> <p>GetValue("cmi.comments_from_learner._count")</p>
cmi.comments_from_learner.n.comment	<p>Description : Cet élément décrira les données d'entrée textuelles [1RTE]. La valeur d'une chaîne de caractères représente la chaîne de caractères localisés.</p> <p>Par exemple :</p> <p>GetValue("cmi.comments_from_learner.0.comment") SetValue("cmi.comments_from_learner.0.comment","Some</p>

	comments about the SCO")
cmi.comments_from_learner.n.location	<p>Description : Cet élément indique le point dans le SCO où le commentaire s'applique. Si aucune valeur n'est indiquée pour l'endroit, alors le commentaire est applicable au SCO entier (dans l'ensemble)</p> <p>Par exemple :</p> <pre>GetValue("cmi.comments_from_learner.0.location") SetValue("cmi.comments_from_learner.0.location","PAGE1SECTION#3")</pre>
cmi.comments_from_learner.n.timestamp	<p>Description : Cet élément indique le point de temps où le commentaire a été créé ou le plus récemment changé. Les périodes de temps dans : de 1 janvier 1970 au 1 janvier 2038 [1RTE].</p> <p>Par exemple :</p> <ul style="list-style-type: none"> •GetValue("cmi.comments_from_learner.0.timestamp") •SetValue("cmi.comments_from_learner.0.timestamp","2003-07-25T03:00:00")
Commentaire du LMS	
cmi.comments_from_lms_children	<p>Description : Cet élément représente une liste des éléments soutenus de modèle de données. Il est typiquement employé par un SCO pour déterminer quels éléments de modèle de données sont soutenus par le LMS. Une chaîne de caractères retournée peut être employé par le SCO pour établir dynamiquement des paramètres pour les demandes de GetValue () et de SetValue().</p> <p>Par exemple :</p> <pre>GetValue("cmi.comments_from_lms_children")</pre>
cmi.comments_from_lms_count	<p>Description : Ce mot-clé est employé pour décrire le nombre courant de commentaires des LMS qui sont actuellement stockés par le LMS. Le nombre au total sera retourné</p>

	<p>Par exemple :</p> <p>GetValue("cmi.comments_from_lms._count")</p>
<p>cmi.comments_from_lms.n.comment</p>	<p>Description : Cet élément décrira des commentaires ou des annotations liés à un SCO [1RTE]. La valeur d'une chaîne de caractères représente la chaîne de caractères localisée.</p> <p>Par exemple :</p> <p>GetValue("cmi.comments_from_lms.0.comment")</p>
<p>cmi.comments_from_lms.n.location</p>	<p>Description : Cet élément indique le point dans le SCO où le commentaire s'applique. Si aucune valeur n'est indiquée pour l'endroit, alors le commentaire est applicable au SCO entier (dans l'ensemble) [1RTE].</p> <p>Par exemple :</p> <p>GetValue("cmi.comments_from_lms.0.location")</p>
<p>cmi.comments_from_lms.n.timestamp</p>	<p>Description : Il semble l'élément cmi.comments_from_learner.n.timestamp (Voyez-vous à la section cmi.comments_from_learner.n.timestamp dessus)</p> <p>Par exemple :</p> <p>GetValue("cmi.comments_from_lms.0.timestamp")</p>
<p>Statut d'accomplissement</p>	
<p>cmi.completion_status</p>	<p>Description : L'IEEE définit quatre valeurs d'état. SCORM lie ces valeurs d'état à la marque restreinte suivante de vocabulaire:</p> <ul style="list-style-type: none"> ▪ "completed ": L'étudiant a éprouvé assez expérience pour considérer le SCO complet [1RTE]. ▪ "incomplete ": L'étudiant n'a pas éprouvé assez de l'expérience pour considérer le SCO complet. ▪ "not attempted ": L'étudiant est considéré comme il n'a jamais employé le SCO. ▪ "unknown ": Aucune affirmation n'est faite. Il indique qu'on ne peut faire aucune affirmation applicable qui indique le statut d'accomplissement. <p>Par exemple :</p> <p>GetValue("cmi.completion_status")</p>

	SetValue("cmi.completion_status","incomplete")
Seuil d'accomplissement	
cmi.completion_threshold	<p>Description : La valeur stockée dans cet élément peut être employée pour déterminer si le SCO devrait être considéré complet. Ceci peut être accompli en comparant le cmi.completion_threshold au cmi.progress_measure, fait par l'étudiant, vers l'accomplissement du SCO.</p> <p>Par exemple :</p> <p>GetValue("cmi.completion_threshold")</p>
Crédit	
cmi.credit	<p>Description : Cet élément indique si l'étudiant sera crédité pour l'exécution dans le SCO.</p> <p>Par exemple :</p> <p>GetValue("cmi.credit")</p>
Entrée	
cmi.entry	<p>Description : Cet élément contient l'information qui affirme que l'étudiant a précédemment consulté le SCO.</p> <p>Par exemple :</p> <p>GetValue("cmi.entry")</p>
Sortie	
cmi.exit	<p>Description : Cet élément indique comment ou pourquoi l'étudiant a laissé le SCO. Cette valeur est employée pour indiquer la raison que le SCO a été sorti</p> <p>Par exemple :</p> <p>SetValue("cmi.exit","suspend").</p>
Interactions	
cmi.interactions._count	<p>Description : Cet mot-clé est employé pour décrire le nombre courant d'interactions stocké par le LMS. Tout le nombre au total sera retourné.</p> <p>Par exemple :</p>

	GetValue ("cmi.ininteractions._count")
cmi.interactions.n.id	<p>Description : Cet élément est une étiquette pour l'interaction.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> GetValue("cmi.interactions.0.id") SetValue("cmi.interactions.0.id","obj1")
cmi.interactions.n.type	<p>Description : Cet élément indique quel type de interaction est enregistré dans un exemple d'une interaction. On l'emploie également pour déterminer comment la réponse d'interaction devrait être interprétée</p> <p>Par exemple :</p> <ul style="list-style-type: none"> GetValue("cmi.interactions.0.type") SetValue("cmi.interactions.0.type","true-false")
cmi.interactions.n.objectives._count	<p>Description : Ce mot-clé de est employé pour décrire le nombre courant d'objectifs stocké par le LMS. Tout le nombre au total sera retourné.</p> <p>Par exemple :</p> <ul style="list-style-type: none"> GetValue("cmi.interactions.0.objectives._count")
cmi.interactions.n.objectives.n.id	<p>Description : Cet élément est une étiquette pour des objectifs liés à l'interaction. L'étiquette sera unique au moins dans la portée du SCO</p> <p>Par exemple :</p> <ul style="list-style-type: none"> GetValue("cmi.interactions.0.objectives.0.id") SetValue("cmi.interactions.0.objectives.0.id","urn:ADL:objective-id-0001")
cmi.interactions.n.timestamp	<p>Description : Cet élément est le point du temps où l'interaction a été rendue la première fois par étudiant pour interaction et réponse</p> <p>Par exemple :</p> <ul style="list-style-type: none"> GetValue("cmi.interactions.0.timestamp") SetValue("cmi.interactions.0.timestamp"2003-07-25T03:00:00")

cmi.interactions.n.correct_responses._count	<p>Description : Ce mot-clé est employé pour décrire le nombre courant de réponses correctes stocké par le LMS. Tout le nombre au total sera retourné</p> <p>Par exemple :</p> <p>GetValue("cmi.interactions.0.correct_responses._count")</p>
cmi.interactions.n.weighting	<p>Description : Cet élément est le poids (weight) donné à l'interaction qui peut être employée par le SCO pour calculer une valeur pour des points. Comment cette valeur ou le calcul des points totaux sont calculés par le SCO, ceci est en dehors de la portée de SCORM</p> <p>Par exemple :</p> <p>GetValue("cmi.interactions.0.weighting") SetValue("cmi.interactions.0.weighting","1.0")</p>
Lentement des Données	
cmi.launch_data	<p>Description : Pendant l'expérience d'apprentissage, il a besoin de fournir au SCO lié à la activité d'apprentissage de l'information de lancement. Cette information ne peut pas être représentée en utilisant des paramètres donc l'élément de modèle de données de cmi.launch_data fournit cette information.</p> <p>Par exemple :</p> <p>GetValue("cmi.launch_data")</p>
Identité d'apprenant	
cmi.learner_id	<p>Description : Cet élément identifie l'étudiant au nom dont le SCO a été lancé. Le cmi.learner_id sera unique au moins dans la portée du SCO. Comment il est assigné est en dehors de la portée de SCORM. Il identifie également l'étudiant baser sur les données de LMS</p> <p>Par exemple :</p> <p>GetValue("cmi.learner_id")</p>
Temps permet au maximum	
cmi.max_time_allowed	<p>Description : Cet élément est la quantité de temps accumulé</p>

	<p>on permet au étudiant d'employer un SCO dans l'essai d'apprentissage. Il commence de la première session jusqu'à ce que l'activité se termine.</p> <p>Par exemple :</p> <p>GetValue("cmi.max_time_allowed")</p>
Mode	
<p>cmi.mode</p>	<p>Description : Cet élément identifie un de trois modes possibles que le SCO peut être présenté à l'étudiant.</p> <ul style="list-style-type: none"> ▪ "browse" : Le SCO est présenté sans intention d'enregistrer n'importe quelles informations sur la session courante d'étudiant ▪ "normal": Le SCO est présenté avec l'intention d'enregistrer l'informations de la session courante d'étudiant. Il est la mode par défaut ▪ "review" : Le SCO a précédemment enregistré des informations sur l'essai d'apprentissage et Il est présenté sans intention de mettre à jour cette information avec des données de la session courante d'étudiant <p>Par exemple :</p> <p>GetValue("cmi.mode")</p>
Score	
<p>cmi.score._children</p>	<p>Description : Cet élément représente une liste des éléments soutenus de modèle de données. Il est typiquement employé par un SCO pour déterminer quels éléments de modèle de données sont soutenus par le LMS. Une chaîne de caractères retourné peut être employé par le SCO pour établir dynamiquement des paramètres pour les demandes de GetValue() et de SetValue().</p> <p>Par exemple :</p> <p>GetValue("cmi.score._children")</p>
<p>cmi.score.raw</p>	<p>Description : Cet élément est un nombre qui reflète l'exécution de l'étudiant relativement à la gamme des valeurs du minimum et du maximum</p> <p>Par exemple :</p> <p>GetValue("cmi.score.raw")</p>

	SetValue("cmi.score.raw","75.0033") SetValue("cmi.score.raw","0.75")
cmi.score.max	<p>Description : C'est la valeur au maximum pour cmi.score.raw</p> <p>Par exemple :</p> GetValue("cmi.score.max") SetValue("cmi.score.max","1.0") SetValue("cmi.score.max","500")
cmi.score.min	<p>Description : C'est la valeur au minimum pour cmi.score.raw</p> <p>Par exemple :</p> GetValue("cmi.score.min") • SetValue("cmi.score.min","1.0") • SetValue("cmi.score.min","500")
Temps de Session	
cmi.session_time	<p>Description : Cet élément est la quantité de temps que l'étudiant a passée en session courante d'étudiant pour ce SCO. Si aucune session d'étudiant n'est en marche, alors le temps de session est le temps où l'étudiant a passé en dernière session d'étudiant pour ce SCO</p> <p>Par exemple :</p> SetValue("cmi.session_time","PT1H5M")
Statut du succès	
cmi.success_status	<p>Description : Cet élément indique si l'étudiant a maîtrisé le SCO. Comment le SCO détermine son cmi.success_status est en dehors de la portée de SCORM. Le SCO a pu baser cette décision sur un certain pourcentage des interactions étant passées, un certain pourcentage des objectifs étant réuni, des points totaux pour un examen etc. Cette valeur indique le statut global de succès pour le SCO.</p> <p>Par exemple :</p> GetValue("cmi.success_status") SetValue("cmi.success_status","passed")

Données suspendues	
cmi.suspend_data	<p>Description : Pendant une expérience d'apprentissage, l'étudiant ou le SCO peut souhaiter suspendre l'essai d'apprentissage sur le SCO et reprendre l'essai d'apprentissage plus tard. Il est de la responsabilité de SCO's de fournir un certain mécanisme pour l'étudiant peut suspendre l'essai courante. Si l'essai d'apprentissage sur le SCO est suspendu, l'état des données d'exécution de SCO persistera jusqu'à la prochaine session d'étudiant sur ce SCO (si le cmi.exit est placé à "suspend"). L'élément cmi.suspend_data fournit l'espace additionnel pour stocker et récupérer des données entre les sessions d'étudiant; les données suspendus peuvent être employé par le SCO pour reprendre la l'essai d'apprentissage</p> <p>Par exemple :</p> <pre>SetValue("cmi.suspend_data","<data><intID>1001</intID> <ans>A</ans></data>") • SetValue("cmi.suspend_data","A1;B2;C11-3")</pre>
Time Limit Action	
cmi.time_limit_action	<p>Description : Cet élément indique que le SCO devrait faire quand cmi.max_time_allowed est excédé</p> <p>Par exemple :</p> <pre>GetValue("cmi.time_limit_action")</pre>
Temps au total	
cmi.total_time	<p>Description : La valeur de cet élément est la somme de tous les temps de la session de l'étudiant (cmi.session_time) accumulé dans l'essai courante d'étudiant. Cet élément est employé pour détecter tout le moment passé en tout des sessions de l'étudiant.</p> <p>Par exemple : <code>GetValue("cmi.total_time")</code></p>