

École polytechnique de Louvain

Analyse bibliographique

Auteur: **Jacques YAKOUB**

Année académique 2019–2020
Master [120] en sciences informatiques

TABLE DES FIGURES

2.1	Les 15 éléments du Dublin Core	4
2.2	Les 9 catégories d'éléments du LOM	5
2.3	Représentation hiérarchique du LOM	6
2.4	Correspondance des éléments entre le Dublin Core et le LOM	6
2.5	MLR - Spécification d'un élément de données	8
2.6	MLR - Exemple de spécification d'un élément de données	8
2.7	Représentation simplifiée du MLR	9
3.1	Exemple de taxonomie pour les exercices de programmation	10
3.2	Liste des sous-domaines de l'informatique selon le CC2005	11
3.3	Exemple de hiérarchie de compétences	12
3.4	Catégories d'exercices informatiques	12

LISTE DES TABLEAUX

2.1	Les 5 types de données du LOM	7
4.1	Éléments de base de notre norme	14
4.2	Éléments d'un mot-clé de notre norme	15
4.3	Quelques éléments additionnels de notre norme	15

TABLE DES EXTRAITS DE CODE

4.1	Exemple d'une ressource information avec notre norme	16
-----	--	----

Après avoir délimité la problématique de notre mémoire (à savoir le référencement et recherche des ressources informatiques¹), nous avons parcouru la toile à la recherche de divers travaux à ce sujet pour tenter d'apporter des réponses à celle-ci. En effet, bien qu'il soit peu probable d'y trouver une solution prête à l'emploi pour couvrir la totalité de la problématique, cette étape n'en demeure pas moins essentielle dans la mesure où des éléments connexes (voire des approches analogues dans d'autres domaines) à notre recherche peuvent émerger et nous permettre de formuler judicieusement notre réponse.

Afin de synthétiser et vous présenter les résultats de notre recherche, nous utiliserons ce fil rouge :

- l'étude de diverses normes pour décrire une ressource
- la recherche de taxonomie pour classer des ressources informatiques

Enfin, nous concluons la présente recherche par les enseignements qu'elle nous a apportés sur notre mémoire par l'élaboration de notre propre norme pour établir une solution équilibrée.

¹Une ressource informatique désigne tout matériel ayant pour attrait le domaine de l'informatique. Exemple : des exercices de programmation, des tutoriels en informatique, des algorithmes, des slides ...

Se poser la question de la représentation d'une ressource est loin d'être anodin dans la mesure où celle-ci doit faire l'objet d'une démarche scientifique valable pour être recevable :

- Quels sont les critères/caractéristiques à notre disposition pour la décrire ?
- Quelle définition (claire et précise) associer à chaque terme spécifique à celle-ci ?

Notre recherche a fait émerger de très nombreuses normes utilisées par le monde pour décrire des ressources éducatives (informatiques ou non). Bien qu'il serait instructif d'étudier l'ensemble des normes existantes pour en déduire les similitudes et différences entre elles, cela peut s'avérer être une tâche chronophage avec une utilité relativement limitée. En effet, nous avons remarqué les faits suivants :

- Certaines normes ne sont que des versions spécialisées d'une autre (par exemple, le **LOMFR** n'est qu'une déclinaison du **LOM**)
- Certaines normes étendent simplement le travail réalisé par une autre norme, en apportant un nombre restreint de changements (comme la **IMS**¹ qui a commencé son développement au départ du **LOM**)
- La norme la plus ancienne que nous avons trouvée date de 1995, ce qui nous indique que la réflexion sur le référencement sur ce type de ressources est relativement récente par rapport à d'autres domaines. À titre informatif, en 1973, il y avait déjà la norme ISO 2709 pour décrire un objet (livre y compris) dans une bibliothèque.

De ce fait, nous n'avons donc retenu ici que les 3 les plus reconnues au niveau mondial (c.-à-d. celles ayant été validées par des organisations fiables comme l'ISO² et la IEEE³) et intéressantes à présenter. Voici donc la liste des normes choisies, ordonnée en fonction de l'année de leur première version, ainsi que l'abréviation en gras que nous utiliserons dès maintenant pour les désigner respectivement :

- la "Dublin Core" (**DC**), officiellement connue sous le nom de "*ISO 15836*"
- la "Learning Object Metadata" (**LOM**), officiellement connue sous le nom de "*IEEE 1484.12.1 – 2002 Standard for Learning Object Metadata*"
- la "Metadata for learning resources" (**MLR**), officiellement connue sous le nom de "*ISO/IEC 19788 Information technology – Learning, education and training – Metadata for learning resources*"

¹<http://www.imsproject.org/>

²International Organization for Standardization - <https://www.iso.org/fr/home.html>

³Institute of Electrical and Electronics Engineers - <https://www.ieee.org/>

Nous allons débiter par expliquer le **DC** dont l'organisation **DCMI** [7] en est l'agence de maintenance. Comme expliqué par la Bibliothèque nationale de France [5], cette norme consiste principalement en "15 éléments tous facultatifs et tous répétables" (que l'on peut classer dans 3 grandes catégories de description : contenu, propriété intellectuelle et l'instance de la ressource). Vous pouvez trouver ci-dessous une illustration expliquant chacun de ces éléments (le terme entre guillemets est le terme d'origine et sa traduction française est entre parenthèses) :

<contributor> (contributeur)	Contributeur (auteur secondaire) du contenu de la ressource : personne physique ou collectivité. [On peut conseiller de saisir cet élément de façon normalisée].
<coverage> (couverture)	Couverture spatio-temporelle ou juridictionnelle du contenu de la ressource. Le contenu de cet élément est, le plus souvent, un nom de lieu et/ou des dates en utilisant un vocabulaire contrôlé : par exemple « France » ; « 19 ^e siècle »...
<creator> (createur)	Créateur du contenu de la ressource : personne physique ou collectivité. [On peut conseiller de saisir cet élément de façon normalisée].
<date> (date)	Date de la création du document. Il est fortement conseillé d'utiliser la norme ISO 8601 qui code la date sous la forme AAAA-MM-JJ. Il est possible de « raffiner » cet élément pour indiquer une date de modification, de sélection de la ressource...
<description> (description)	Description du contenu de la ressource. Le contenu de cet élément peut être un résumé, une table des matières ou encore une description en texte libre.
<format> (format)	Manifestation physique ou numérique de la ressource. Le contenu de cet élément est, le plus souvent, le support ou/et le type d'application informatique nécessaire à sa lecture, par exemple image/jpeg ou text/pdf. On peut également donner des indications concernant la taille du document (par exemple nombre d'octets). L'utilisation de la liste MIME (liste des « types de média définis sur Internet ») est recommandée.
<identifier> (identifiant)	Référence unique et non ambiguë de la ressource. Il est recommandé d'utiliser un système d'identification, comme URI (<i>Uniform Resource Identifier</i>) ou URL (<i>Uniform Resource Locator</i>).
<language> (langue)	Langue du contenu de la ressource. L'utilisation de la norme ISO 639 est recommandée.
<publisher> (éditeur)	Éditeur (au sens commercial) de la ressource : personne physique ou collectivité.
<relation> (relation)	Ressource(s) liée(s) à la ressource décrite (autres versions, formats...). L'utilisation d'un système d'identification (comme URI, URL...) est recommandée.
<rights> (droits)	Informations sur les droits concernant la ressource décrite (gestion des droits, droits d'auteur...).
<source> (source)	Référence d'origine de la ressource décrite. L'utilisation d'un système d'identification est recommandée.
<subject> (sujet)	Sujet du document. L'utilisation d'un système de mots-clefs ou d'indices de classification est recommandée, en particulier les classifications reconnues ou les vocabulaires contrôlés.
<title> (titre)	Titre principal de la ressource, le plus souvent donné par son créateur. [Même s'il n'en est pas question dans la norme, on peut recommander de rendre cet élément obligatoire pour la bonne gestion des systèmes et des échanges de données.]
<type> (type)	Information sur la nature ou le genre de la ressource (texte, image...). L'utilisation d'un terme choisi dans une liste contrôlée, par exemple celle proposée sur le site Dublin Core, est recommandée.

FIGURE 2.1 – Les 15 éléments du Dublin Core [18]

Le lecteur/lectrice attentif/attentive que vous êtes aura noté la présence du terme "vocabulaire contrôlé" [22] ("controlled vocabulary" dans le texte d'origine de la norme [7]). Comme l'explique Wikipédia [22], il s'agit d'un "lexique dont le but est de rendre possible l'organisation des connaissances afin d'optimiser la recherche d'information." Étant donné qu'il impose l'usage de termes prédéfinis, celui-ci s'oppose au "langage naturel, où aucune restriction n'existe quant au choix du vocabulaire". Illustrons ceci par un petit exemple sur l'élément "format" avec la même information, mais exprimée différemment :

- vocabulaire contrôlé (par exemple : MIME⁴) : application/pdf;charset=utf-8
- langage naturel : fichier PDF en UTF-8

La norme **DC** n'impose pas des normes précises (mais recommande explicitement [7] l'usage des "vocabulaires contrôlés" [22]) pour les divers éléments expliqués ci-dessous : en effet, des organisations telles que la Bibliothèque nationale de France [5] se doivent d'avoir des exigences strictes de référencement pour optimiser la recherche d'informations.

Enfin, la norme **DC** inclut la notion de "profil d'application" (de son nom technique "Dublin Core™ Application Profile" [8]). Derrière ce nom se cache un constat assez simple à établir : tenter d'établir une norme unique et stricte est une procédure qui ne peut aboutir. En effet, chaque organisation a des besoins et des exigences différents sur les divers éléments de la norme **DC**. Comme expliqué par la **DCMI** [8], le profil d'application consiste donc en une déclaration qui spécifie quels éléments de la norme **DC** et quelles normes supplémentaires/spécifiques dessus sont d'application.

Nous allons poursuivre par le **LOM** dont le comité **IEEE-LTSC-LOM** [4] (dépendant donc du IEEE que nous avons expliqué précédemment) est responsable. Utilisant le **DC** comme point de départ, celle-ci propose comme expliqué par Wikipédia [20] et "9 catégories regroupant 68 éléments" dont voici un petit descriptif :

Description des catégories du LOM	
Catégories	Description
1. General	Regroupe les caractéristiques d'une ressource qui sont indépendantes du contexte d'utilisation (titre, langue, description, etc.).
2. Life cycle	Décrit l'état actuel d'une ressource et qui y a contribué durant son évolution (version, entités contributives, etc.).
3. Meta-Metadata	Rassemble les données détaillant la fiche descriptive elle-même plutôt qu'une ressource.
4. Technical	Assemble les caractéristiques techniques d'une ressource (format, taille, localisation...).
5. Educational	Décrit les caractéristiques pédagogiques d'une ressource (type de ressource, rôle de l'utilisateur, contexte d'utilisation...).
6. Rights	Spécifie les conditions d'utilisation d'une ressource (coûts, droits d'auteur , etc.).
7. Relation	Décrit, s'il y a lieu, la relation entre une ressource et d'autres ressources.
8. Annotation	Permet des commentaires sur l'utilisation pédagogique d'une ressource.
9. Classification	Décrit la localisation d'une ressource dans un certain système de classification.

FIGURE 2.2 – Les 9 catégories d'éléments du **LOM** [20]

Nous vous proposons dès lors une illustration pour représenter concrètement les divers éléments du **LOM** dans ces catégories dans la figure 2.3.

⁴RFC 6838 - <https://tools.ietf.org/html/rfc6838>

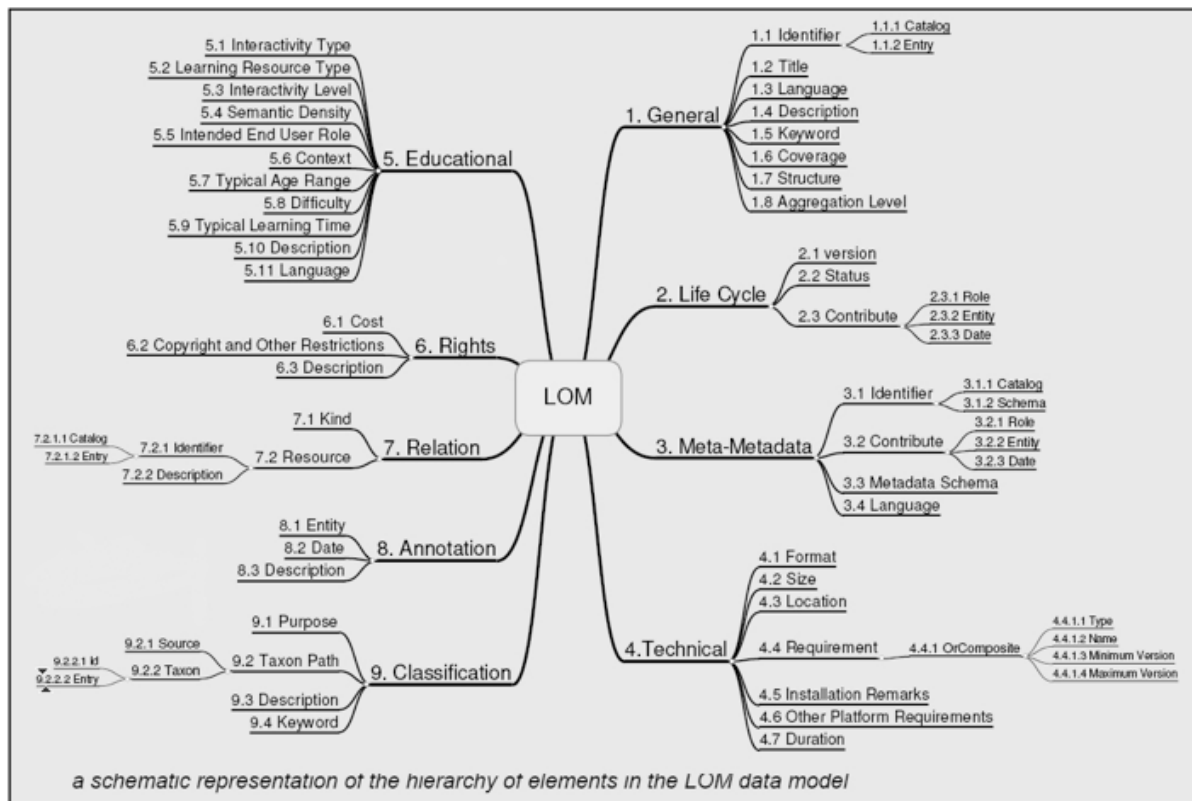


FIGURE 2.3 – Représentation hiérarchique du LOM [10]

Cette norme, qui dispose aussi des profils d'application que nous avons expliqués ci-dessus, a aussi prévu un processus de correspondance entre chaque élément du **DC** vers son équivalent du **LOM** afin de faciliter la transition vers celle-ci.

<i>Dublin Core</i>	<i>IEEE LOM</i>	
Data Element	Identifier	Data Element
DC.Identifier	1.1.2	General.Identifier.Entry
DC.Title	1.2	General.Title
DC.Language	1.3	General.Language
DC.Description	1.4	General.Description
DC.Subject	1.5	General.Keyword
DC.Coverage	1.6	General.Coverage
DC.Type	5.2	Educational.LearningResourceType
DC.Date	2.3.3	LifeCycle.Contribute.Date
DC.Creator	2.3.2	LifeCycle.Contribute.Entity
DC.OtherContributor	2.3.2	LifeCycle.Contribute.Entity
DC.Publisher	2.3.2	LifeCycle.Contribute.Entity
DC.Format	4.1	Technical.Format
DC.Rights	6.3	Rights.Description
DC.Relation	7.2.2	Relation.Resource.Description
DC.Source	7.2	Relation.Resource

FIGURE 2.4 – Correspondance des éléments entre le Dublin Core et le LOM [9]

Enfin, tout comme la norme **DC**, le **LOM** ne fait que recommander de bonnes pratiques. Une des recommandations spécifiques du **LOM** est de préciser un type de donnée spécifique pour chaque élément du **LOM**. En effet, dans le **DC**, tous les éléments étaient encodés soit dans le langage naturel soit dans un vocabulaire contrôlé. Cette liberté rend l'indexation et recherche de ressources très difficile à gérer. Vous pouvez trouver ci-dessous une explication sur ces 5 types de données :

Type de donnée	But	Exemple
CharacterString (CS)	Valeur servant d'identifiant formel ou de nom	application/pdf
DateTime (DT)	Exprime la date et l'heure sous une forme prescrite et formatée	2001-12-31T12 :59 :60,999+02 :30
Duration (DU)	Exprime la durée sous une forme normalisée et formatée	P2Y1M2DT1H20M25.55S
LangString (LS)	Exprime une combinaison d'un indicateur de langue et d'une chaîne de caractères formulée dans celle-ci	FR -
Vocabulary (Vs, Ve)	Exprime le besoin d'un vocabulaire contrôlé ou à des listes prédéterminées de termes ou de nombres	Très faible, faible, moyen, élevé, très élevé

TABLE 2.1 – Les 5 types de données du **LOM** [20]

Nous allons conclure par le **MLR** dont l'**ISO/IEC JTC1** (dépendant de l'ISO que nous avons expliqué précédemment) est responsable. Cette norme a pour vocation à unifier **DC** et **LOM** (et possède donc un processus de correspondance permettant de migrer du **LOM** au **MLR**). Comme expliqué par Wikipédia [19], les normes **DC** et **LOM** ont plusieurs problèmes en matière d'interopérabilité ⁵ :

- Les bonnes pratiques de convention sont uniquement recommandées et de ce fait pas obligatoires. (par exemple, un élément de type date en **DC** peut être libellé de manière intraitable pour des recherches)
- L'existence de définitions ambiguës et donc soumises à notre subjectivité personnelle. Par exemple :

⁵(Définition Larousse) <https://www.larousse.fr/dictionnaires/francais/interopérabilité/43787>

- Dans le **DC**, l'élément date peut être lié à une date de sa création ou de sa publication selon le contexte.
- Dans le **LOM**, la multitude des profils d'application fait émerger des vocabulaires contrôlés parfois difficilement conciliables entre eux.

De ce fait, le **MLR** tente de résoudre ces problèmes par le biais des "Data Element Specification (DES)" [19] dont vous pouvez trouver des explications détaillées sur chaque champ par les ressources que nous avons utilisées ([1] et [19]), illustrés par un exemple :

Spécification d'un élément de données (DES)	
Identifiant (obligatoire)	
Attributs de l'élément de données	
Nom de la propriété (obligatoire)	
Définition (obligatoire)	
Indicateur linguistique (obligatoire)	
Domaine (obligatoire)	
Image (obligatoire)	
Règles de contenu (conditionnel)	
Raffine (conditionnel)	
Exemple(s) (facultatif)	
Note(s) (facultatif)	

FIGURE 2.5 – MLR - Spécification d'un élément de données [1]

Spécification d'un élément de données (DES)	
Identifiant	ISO_IEC_19788-3:2010::DES0300
Attributs de l'élément de données	
Nom de la propriété	format (eng) format (fra)
Définition	file format, physical medium, or dimensions of the learning resource (eng) format de fichier, type de média ou dimensions de la ressource pédagogique (fra)
Indicateur linguistique	non-linguistique
Domaine	ISO_IEC_19788-1:2010::RC0002 (Ressource d'apprentissage)
Image	Littéral
Règles de contenu	RS_DES0004
Raffine	ISO_IEC_19788-2:2010::DES0900 (format)
Exemple(s)	video/mpeg text/html
Note(s)	-

FIGURE 2.6 – MLR - Exemple de spécification d'un élément de données [1]

Si cette norme peut paraître complexe à comprendre par rapport aux normes précédentes, notre recherche bibliographique nous a permis de trouver divers documents pour résorber ce problème dont notamment ceux du doctorant Nikos Palavitsinis ([13] , [12] ainsi que sa présentation Slideshare [11]) :

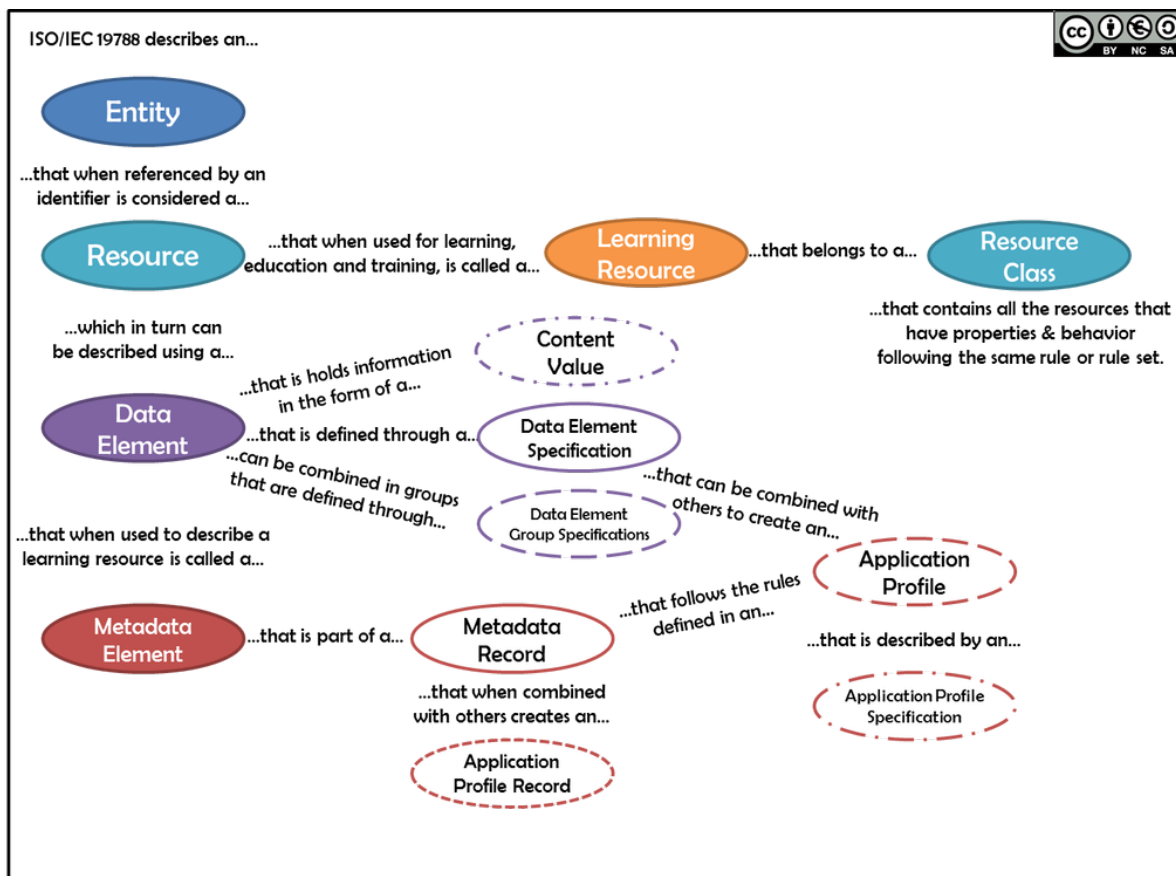


FIGURE 2.7 – Représentation simplifiée du MLR [13]

Pour conclure ce chapitre, nous avons retiré des enseignements de cette étude des normes, qui nous guideront notamment au chapitre 4 :

- Beaucoup d'éléments dans les diverses normes étudiées sont optionnels. De ce fait, certains éléments sont plus appropriés à un contexte que pour un autre.
- Quelle que soit la classification choisie des éléments, il convient d'utiliser quand cela est possible des vocabulaires contrôlés pour répondre à des besoins de recherche dans les ressources.
- Un soin tout particulier doit être accordé à la rédaction des différents éléments d'une norme. Une mauvaise interprétation peut conduire à un nivellement par le bas des objectifs à atteindre.

Dans le chapitre suivant, nous allons nous intéresser aux manières possibles de classer des ressources informatiques puisque notre recherche n'a pas fait émerger une quelconque norme spécialement prévue pour ce contexte.

Après avoir abordé la représentation d'une ressource, il convient d'aborder la question de la taxonomie¹ de l'informatique. De manière similaire au chapitre précédent, des interrogations analogiques peuvent se poser :

- Quelles "catégories"² avec leurs caractéristiques/valeurs propres devons-nous utiliser ?
- Quelle définition (claire et précise) associer à chaque terme ?

Notre recherche n'a pas permis de trouver un consensus suffisamment partagé au niveau mondial sur la taxonomie. En revanche, un début de réponse se trouve dans les normes étudiées précédemment telles que le **LOM** (cf. les figures 2.2 et 2.3). En effet, elles permettent de catégoriser une ressource, qu'elle soit informatique ou non. Le **LOM** permet en outre, par le biais de la catégorie n°9, de décrire la ressource dans un système de classification (propre à chaque organisation), tel que celui ci-dessous :

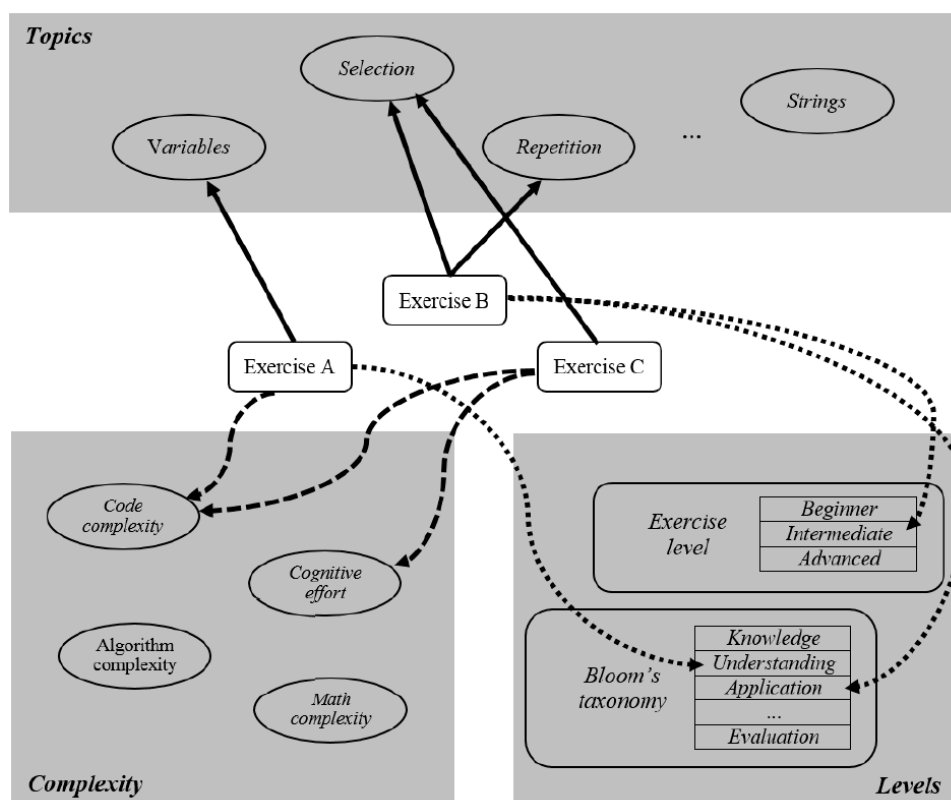


FIGURE 3.1 – Exemple de taxonomie pour les exercices de programmation [14]

¹Comme expliqué par Wikipédia[21], ce terme désigne une "méthode de classification des informations dans une architecture structurée de manière évolutive".

²Le vrai terme dans la littérature scientifique est "taxon" que nous invitons à découvrir par vos propres moyens

Comme expliqué par un rapport [2], l'informatique est une discipline qui se divise en plusieurs (sous) domaines/sujets que vous retrouverez ci-dessous. Bien que cette liste soit l'aboutissement de longues années de travail, elle ne doit pas être considérée comme définitive, car comme l'illustrent ces articles ([17] et [2]), de nouveaux domaines (ou des définitions modifiées) peuvent émerger au fur et à mesure que le temps passe, notamment avec l'arrivée du **CC2005** [15] (Computing Curricula 2005).

Knowledge Area
Programming Fundamentals
Integrative Programming
Algorithms and Complexity
Computer Architecture and Organization
Operating Systems Principles & Design
Operating Systems Configuration & Use
Net Centric Principles and Design
Net Centric Use and configuration
Platform technologies
Theory of Programming Languages
Human-Computer Interaction
Graphics and Visualization
Intelligent Systems (AI)
Information Management (DB) Theory
Information Management (DB) Practice
Scientific computing (Numerical mthds)
Legal / Professional / Ethics / Society
Information Systems Development
Analysis of Business Requirements
E-business
Analysis of Technical Requirements
Engineering Foundations for SW
Engineering Economics for SW
Software Modeling and Analysis
Software Design
Software Verification and Validation
Software Evolution (maintenance)
Software Process
Software Quality
Comp Systems Engineering
Digital logic
Embedded Systems
Distributed Systems
Security: issues and principles
Security: implementation and mgt
Systems administration
Management of Info Systems Org.
Systems integration
Digital media development
Technical support

FIGURE 3.2 – Liste des sous-domaines de l'informatique selon le **CC2005** [3]

Un fait important relevé par un rapport [2] est que l'informatique se concentre autour de 3 processus de base (théorie, abstraction et conception). Ceci pourrait permettre d'avoir un critère supplémentaire pour classer des ressources informatiques au lieu d'inférer cela par le contexte.

Cette recherche bibliographique sur les ressources informatiques a également révélé des tentatives précédentes de classification de compétences : le **CC2005** [15] que nous avons évoquée plus tôt (utilisable par exemple conjointement avec une norme [3] dérivée du **DC** que nous avons trouvé) ou des propositions de modèles dont notamment celle que nous avons trouvé dans un article de recherche [6] qui proposait de décomposer le concept de compétence en principalement 3 termes connexes, connus sous l'anagramme **KSD** ("**K**nowledge, **S**kill, et **D**isposition" que l'on pourrait respectivement traduire en français, en respectant le sens exprimé de l'article, par "savoirs théoriques, savoirs pratiques et dispositions") dont vous pourrez trouver une mise en situation en figure 3.3.

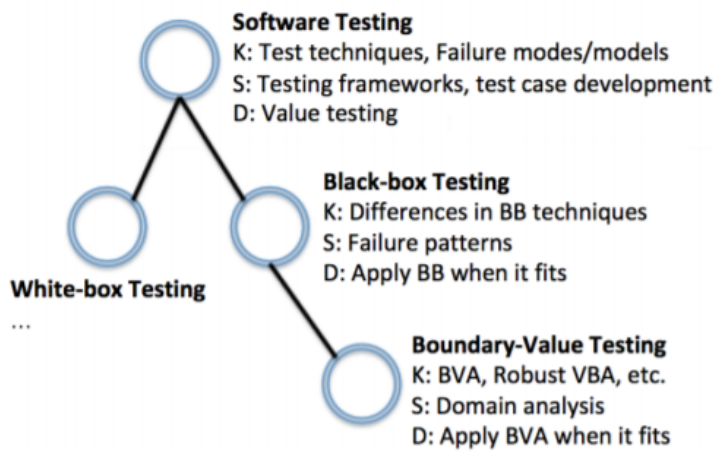


FIGURE 3.3 – Exemple de hiérarchie de compétences [6]

Enfin, dans le cas spécifique d'exercice de programmation, il convient d'avoir une idée du type d'exercice sans forcément lire sa description pour comprendre de quoi il en retourne. Ceci permet en autres de favoriser une diversification des compétences. L'analyse a fait surgir un document récent qui synthétise les plus fréquemment rencontrés :

Concept	Description
Blank sheet	This kind of exercise provides a blank sheet for the student to write his/her solution source code from the scratch.
Code extension	This kind of exercise provides partially finished solution source code (the provided parts are not subject to change by the student) which the student has to complete.
Code improvement	This kind of exercise provides correct initial source code which does not yet achieve all the goals specified in the exercise specification, so the student has to modify it to solve the exercise.
Buggy code	This kind of exercise provides code with bugs (and failed tests) to foster the student to find the right code.
Fill-in the gap	This kind of exercise provides code with missing parts and asks students to fill them with the right code.
Mixed code	This kind of exercise breaks a solution into several blocks of code, mixes them, and asks students to sort them (e.g., quicksort algorithm).
Show me	This kind of exercise defines a small set of primitives that can be used to solve the challenge (e.g., <code>move('red', 'left')</code> meaning move red block to left stack) and provides a visual animation of the code execution.
Spot the bug	This kind of exercise provides code with bugs and asks students to merely indicate the location of the bugs.

FIGURE 3.4 – Catégories d'exercices informatiques [16]

Malgré tous nos efforts pour trouver une taxonomie cohérente à notre situation, aucune d'entre elles ne semble correspondre à nos besoins spécifiques concernant les ressources informatiques (en plus de ne pas susciter un grand intérêt). Dans notre cas, nous aurions besoin d'une taxonomie laissant la possibilité à chaque organisation de faire ce qu'elles veulent, par l'usage d'un vocabulaire flexible et contrôlé. Ce dernier pourrait ainsi évoluer avec le temps et la plateforme qui l'utilise afin de répondre aux besoins des différents utilisateurs de ce même vocabulaire. Nous évoquons cette piste dans le chapitre 4 en proposant notre propre taxonomie adéquate au domaine des ressources informatiques.

Après étude approfondie de la situation, nous avons jugé utile de créer notre propre norme et terminologie pour les raisons suivantes :

- Les 3 normes que nous avons étudiées (**DC**, **LOM** et **MLR**) proposaient des éléments dits "généralistes" sur des ressources (auteur, copyright, date de création/modification ...) et de ce fait ne tiennent pas compte d'éléments spécifiques des ressources informatiques (langage de programmation, thématique ...). Les quelques normes dérivées de celles-ci, orientées au contexte des ressources informatiques (telles que [3] et [16]), proposent certes des avantages, mais aussi des choix discutables sur certains aspects de leur taxonomie.
- Les "vocabulaires contrôlés" [22] sur les éléments sont au coeur de la problématique. En effet, dépendre exclusivement de champs libres rendra l'indexation et la recherche de ressources de qualité impossible. Comme expliqué au chapitre 2, ce vocabulaire se doit être bien défini et clair or l'analyse bibliographique des différents documents a illustré le fait que certains vocabulaires contrôlés n'étaient pas assez génériques (c.-à-d. trop orientés vers un contexte très particulier), et de ce fait difficilement applicables à d'autres organisations.
- La question de la gestion des vocabulaires contrôlés personnalisés (c.-à-d. créés par nos soins) n'est presque jamais soulevée. La recommandation générale dans les normes étudiées était de prendre une sémantique standardisée telle que certains RFC¹ (comme **RFC 6838**), quand cela est possible. Pour certains éléments (comme le langage de programmation), il n'est pas toujours possible d'avoir une liste exhaustive des valeurs possibles. C'est pourquoi nous avons pris la décision d'avoir des vocabulaires contrôlés "extensibles" (c.-à-d. y associer un statut pour encourager la découverte et la modération de valeurs reconnues, dépréciées ...).

Maintenant que nous avons éclairci ces points, nous pouvons passer aux explications sur notre norme. Celle-ci s'est construite sur base de notre réflexion principalement autour des normes **DC** et **LOM**. Contrairement aux normes étudiées, notre norme a fait le choix du schéma **JSON**² (par opposition au **XML**³), car bien qu'étant moins extensible (notamment sur le nombre limité de types de données) que son rival, celui-ci offre une grande facilité de manipulation et de syntaxe aussi bien pour un être humain qu'une machine. Un aspect très important était de proposer une norme ouverte proposant des lignes directrices adaptées aux visions différentes en termes de taxonomie, que nous allons expliquer à la page suivante. En effet, dans le but de référencer un maximum de ressources de sources différentes en un minimum de temps, des robots d'indexation⁴ seront plus que certainement utilisés et il convient dès lors de proposer une structure en conséquence.

¹Requests For Comments - https://fr.wikipedia.org/wiki/Request_for_comments

²JavaScript Object Notation (RFC-8259)

³Extensible Markup Language - <https://www.w3.org/TR/2006/REC-xml11-20060816/>

⁴En anglais, le terme officiel est crawler

Par similarité avec les autres normes, les termes techniques de notre norme seront en anglais, mais nous les présenterons en français. Débutons par les éléments de base de notre norme :

Élément	Cardinalité	Description
title	1	Il s'agit du titre de la ressource. S'agissant d'un champ libre, nous recommandons de se référer à l'élément "title" du DC ou 1.2 du LOM
description	1	Il s'agit d'une description de la ressource. S'agissant d'un champ libre, nous recommandons de se référer à l'élément "description" du DC ou 1.4 du LOM
createdAt	1	Il s'agit de la date de la création de la ressource. Contrairement au type DT du LOM (que nous avons évoqué dans la table 2.1), on impose l'usage exclusif de la norme ISO 8601 .
updatedAt	1	Il s'agit de la date de la dernière mise à jour de la ressource. Contrairement au type DT du LOM (que nous avons évoqué dans la table 2.1), on impose l'usage exclusif de la norme ISO 8601 .
state	1	Il s'agit de l'état de la ressource dans le système. Nous recommandons l'usage de ce vocabulaire contrôlé : <ul style="list-style-type: none"> • DRAFT : l'état par défaut (par exemple la ressource vient d'être insérée dans un système) • PENDING : la ressource est prête pour validation • VALIDATED : la ressource est validée par un administrateur • NOT_VALIDATED : la ressource est refusée par un administrateur • ARCHIVED : la ressource est archivée
version	1	Il s'agit de la version de la ressource. Contrairement au LOM , nous imposons l'usage d'un type numérique positif pour représenter cette information. (0 : version initiale de la ressource, 1 : première modification de la ressource ...)
url	0-1	URL qui référence la ressource. Cette adresse doit être conforme au RFC 3986 .

TABLE 4.1 – Éléments de base de notre norme

Ensuite, tout comme les normes **DC** et **LOM**, un certain nombre d'éléments facultatifs et répétables sont prévus. Comme nous l'avons déjà expliqué, la taxonomie et terminologie de l'informatique étant propre à chaque organisation, nous devons trouver un moyen de concilier tout ceci (cf. vocabulaires contrôlés "extensibles"). L'idée retenue a été de les consigner sous une clé "tags" représentant un tableau dont chaque cellule est constituée des éléments suivants :

Élément	Description
text	Champ textuel libre. Nous recommandons quand cela est possible de suivre une norme reconnue (par exemple l'ISO 639 pour les langues).
category	Identifiant permettant d'identifier à quel élément le mot-clé se rapporte

state	<p>Il s'agit de l'état du mot-clé dans le système. Nous recommandons l'usage de ce vocabulaire contrôlé :</p> <ul style="list-style-type: none"> • PENDING : l'état par défaut (par exemple le mot-clé est en cours de validation) • NOT_VALIDATED : le mot-clé n'est officiellement pas reconnu • DEPRECATED : le mot-clé devrait plus être utilisé ultérieurement • VALIDATED : le mot-clé est officiellement reconnu
-------	---

TABLE 4.2 – Éléments d'un mot-clé de notre norme

De cette manière, nous proposons une alternative flexible permettant de réutiliser des éléments du **LOM** tout en rajoutant des nouveaux et d'implémenter de manière plus simplifiée une recherche multicritère⁵. Voici quelques éléments récurrents que nous avons pu identifier durant la conception de notre solution :

Identifiant	Élément	Description
2	cours	Il s'agit de décrire à quel(s) domaine(s) de l'informatique (ou cours) la ressource fait attrait. Nous recommandons de partir de la liste des domaines en figure 3.2 avant de créer de nouvelles entrées.
3	type d'exercice	Il s'agit de décrire la catégorie d'une ressource informatique de type "exercice" auquel elle se rapporte. Nous recommandons de partir de la liste en figure 3.4 avant de créer une nouvelle catégorie.
6	auteur	Il s'agit de décrire le(s) auteur(s) de la ressource. Nous recommandons pour ce champ libre d'adopter une convention claire d'écriture (par exemple le prénom suivi d'un espace et puis le nom), mais nous restons tributaires de la coopération du créateur de la ressource.
8	thématique	Il s'agit de décrire le(s) thème(s) liés à la ressource informatique. Puisque nous n'avons trouvé aucun consensus suffisamment partagé sur ce domaine, ce champ libre est impacté par la subjectivité du créateur de la ressource (exemples : "listes chaînées", "pointeurs", "CUnit" ...).
9	mots-clés	Il s'agit de référencer les mots-clés n'entrant dans aucune des catégories d'éléments existantes. Ce champ libre est exclusivement impacté par la subjectivité du créateur de la ressource (exemples : "Module 3", "Mission 42", "Septembre 2016").
10	licence	Il s'agit de décrire à quel(s) license(s) logiciels la ressource est soumise. Nous recommandons de partir des identifiants du standard SPDX ⁶ avant de créer de nouvelles entrées.

TABLE 4.3 – Quelques éléments additionnels de notre norme

⁵Par exemple : "Je veux rechercher toutes les ressources traitant d'intelligence artificielle, créées par l'auteur X ou l'auteur Y, réalisables sur la plateforme Z"

⁶<https://spdx.org/licenses/>

Pour illustrer l'encodage d'une ressource informatique avec notre norme, nous vous proposons l'exemple suivant :

Listing 4.1 – Exemple d'une ressource information avec notre norme

```
1 {
2   "title": "Opérations sur les bits",
3   "description": "...",
4   "url": "https://inginius.info.ucl.ac.be/course/LINGI112
5     2/execute",
6   "createdAt": "2019-12-22T15:18:31.090Z",
7   "updatedAt": "2019-12-22T15:19:33.473Z",
8   "state": "DRAFT",
9   "version": 1,
10  "tags": [
11    {
12      "text": "[LSINF1121] Algorithmique et structures
13        de données",
14      "state": "VALIDATED",
15      "category": 2
16    },
17    {
18      "text": "code",
19      "state": "DEPRECATED",
20      "category": 3
21    },
22    {
23      "text": "Mathieu Xhonneux",
24      "state": "PENDING",
25      "category": 6
26    },
27    {
28      "text": "John Aoga",
29      "state": "PENDING",
30      "category": 6
31    },
32    {
33      "text": "CC-BY-NC-4.0",
34      "state": "VALIDATED",
35      "category": 10
36    }
37  ]
38 }
```

Nous concluons le présent travail en précisant que la présente norme (et la taxonomie liée) possède bien entendu des faiblesses en dépit de sa structure ouverte : comme les efforts précédents que nous avons expliqués aux deux derniers chapitres (**LOM**, la proposition d'arborescence de compétences en figure 3.3 ...), l'internationalisation de la description d'une ressource est le principal problème. En effet, des divergences manifestes de terminologie et/ou de description existent (par exemple la vision de l'enseignement aux États-Unis comparé en Europe) et des normes tentant de résorber ce problème (telles que le **MLR**) ne semblent pas encore suffisamment adoptées et/ou connues.

- [1] Yolaine BOURDA et al. “Métadonnées pour ressources d’apprentissage (MLR) : nouvelle norme ISO de description de ressources pédagogiques”. Dans : *STICEF (Sciences et Technologies de l’Information et de la Communication pour l’Éducation et la Formation)* 17 (2010), 11 pages. URL : <https://hal.archives-ouvertes.fr/hal-00696450>.
- [2] Peter J. DENNING et al. “Computing as a Discipline”. Dans : *IEEE Computer* 22.2 (1989), p. 63-70. DOI : 10.1109/2.19833. URL : <https://doi.org/10.1109/2.19833>.
- [3] Stephen H. EDWARDS et al. “Developing a Common Format for Sharing Programming Assignments”. Dans : *SIGCSE Bull.* 40.4 (nov. 2008), p. 167-182. ISSN : 0097-8418. DOI : 10.1145/1473195.1473240. URL : <https://doi.org/10.1145/1473195.1473240>.
- [4] Institute of ELECTRICAL et Electronics ENGINEERS. “IEEE Standard for Learning Object Metadata”. Dans : *IEEE Std 1484.12.1-2002* (2002), p. 1-40.
- [5] Bibliothèque nationale de FRANCE. *Dublin Core*. URL : <https://www.bnf.fr/fr/dublin-core#bnf-dublin-core-et-bonnes-pratiques>.
- [6] Stephen FREZZA et al. “Modelling competencies for computing education beyond 2020 : a research based approach to defining competencies in the computing disciplines”. Dans : *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE 2018, Larnaca, Cyprus, July 02-04, 2018*. Sous la dir. de Guido RÖSSLING et Bruce SCHARLAU. ACM, 2018, p. 148-174. DOI : 10.1145/3293881.3295782. URL : <https://doi.org/10.1145/3293881.3295782>.
- [7] Dublin Core™ Metadata INITIATIVE. *DCMI Metadata Terms*. URL : <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>.
- [8] Dublin Core™ Metadata INITIATIVE. *Dublin Core™ Application Profile Guidelines*. URL : <https://www.dublincore.org/specifications/dublin-core/application-profile-guidelines/>.
- [9] Pythagoras KARAMPIPERIS et Demetrios SAMPSON. “Enhancing Educational Metadata Management Systems to Support Interoperable Learning Object Repositories”. Dans : jan. 2003, p. 214-218. DOI : 10.1109/ICALT.2003.1215059. URL : https://www.researchgate.net/publication/221425064_Enhancing_Educational_Metadata_Management_Systems_to_Support_Interoperable_Learning_Object_Repositories.
- [10] Dimitrios A. KOUTSOMITROPOULOS et al. *The Use of Metadata for Educational Resources in Digital Repositories : Practices and Perspectives*. URL : <http://www.dlib.org/dlib/january10/kout/01kout.print.html>.
- [11] Nikos PALAVITSINIS. *Making Sense of ISO/IEC 19788*. Juin 2016. URL : <https://www.slideshare.net/nikospala/making-sense-of-isoiec-19788>.

- [12] Nikos PALAVITSINIS. *Making Sense of ISO/IEC 19788 (part I)*. URL : <https://palavitsinis.wordpress.com/2016/06/23/making-sense-of-isoiec-19788-part-i/>.
- [13] Nikos PALAVITSINIS. *Making Sense of ISO/IEC 19788 (part II)*. URL : <https://palavitsinis.wordpress.com/2016/07/04/making-sense-of-isoiec-19788-part-ii/>.
- [14] Álvaro SANTOS, Anabela GOMES et Antonio MENDES. “A taxonomy of exercises to support individual learning paths in initial programming learning”. Dans : oct. 2013, p. 87-93. DOI : 10.1109/FIE.2013.6684794.
- [15] Russell SHACKELFORD et al. “Computing Curricula 2005 : The Overview Report”. Dans : *SIGCSE Bull.* 38.1 (mar. 2006), p. 456-457. ISSN : 0097-8418. DOI : 10.1145/1124706.1121482. URL : <https://doi.org/10.1145/1124706.1121482>.
- [16] Jakub SWACHA et al. “Defining Requirements for a Gamified Programming Exercises Format”. Dans : *Knowledge-Based and Intelligent Information & Engineering Systems : Proceedings of the 23rd International Conference KES-2019, Budapest, Hungary, 4-6 September 2019*. Sous la dir. d’Imre J. RUDAS et al. T. 159. Procedia Computer Science. Elsevier, 2019, p. 2502-2511. DOI : 10.1016/j.procs.2019.09.425. URL : <https://doi.org/10.1016/j.procs.2019.09.425>.
- [17] Allen B. TUCKER et Peter WEGNER. “Computer Science and Engineering : The Discipline and Its Impact”. Dans : *The Computer Science and Engineering Handbook*. Sous la dir. d’Allen B. TUCKER. CRC Press, 1997, p. 1-15.
- [18] Isabelle WESTEEL. “Indexer, structurer, échanger : métadonnées et interopérabilité”. Dans : *OpenEdition Books* (2010), p. 100-123. DOI : <https://doi.org/10.4000/books.pressesenssib.434>. URL : <https://books.openedition.org/pressesenssib/434>.
- [19] WIKIMEDIA. *ISO/IEC 19788*. Avr. 2020. URL : https://en.wikipedia.org/wiki/ISO/IEC_19788.
- [20] WIKIPEDIA. *Learning Object Metadata*. Avr. 2020. URL : https://fr.wikipedia.org/wiki/Learning_Object_Metadata.
- [21] WIKIPEDIA. *Taxonomie (homonymie)*. Fév. 2020. URL : [https://fr.wikipedia.org/wiki/Taxonomie_\(homonymie\)](https://fr.wikipedia.org/wiki/Taxonomie_(homonymie)).
- [22] WIKIPEDIA. *Vocabulaire contrôlé*. Avr. 2020. URL : https://fr.wikipedia.org/wiki/Vocabulaire_contr%C3%B4l%C3%A9.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl