

Le protocole OAI et SDX 2

Réflexions sur l'OAI, le support OAI dans SDX 2,
ainsi que sur l'opportunité de faire des tests OAI

19 novembre 2002

Document rédigé par :



Martin Sévigny (sevigny@ajlsm.com)
Ingénieur documentaire
AJLSM – 17 rue Vital Carles – 33000 Bordeaux

Présentation

L'une des préoccupations principales aujourd'hui en informatique documentaire est l'interopérabilité, c'est-à-dire la possibilité de partager des informations provenant de sources hétérogènes, voire même d'utiliser des services distants sans en connaître nécessairement tous les détails techniques. Plusieurs approches technologiques permettent d'atteindre des niveaux intéressants d'interopérabilité, et toutes doivent adresser quelques aspects centraux, en particulier les vocabulaires de description, les formats de structuration des données descriptives, et enfin les méthodes à utiliser pour échanger ces données. Le projet OAI¹ a été créé dans le but de proposer des solutions d'interopérabilité qui incluent ces différents aspects.

Créer de l'information documentaire, qu'elle soit interopérable ou non, est une activité intéressante, mais en général cet intérêt se manifestera lorsqu'on aura des outils permettant d'exploiter cette information documentaire ; recherche, tri, affichage, extraction, etc. La plate-forme SDX 2² propose justement des outils pour la consultation et la recherche d'information documentaire en format XML. Jusqu'à maintenant, SDX 2 n'a pas été conçu pour rendre les données qu'il gère automatiquement interopérables selon un format ou protocole particulier. Toutefois, avec l'évènement de l'OAI et surtout sa popularité récente, il est intéressant de voir comment SDX 2 peut devenir un outil *interopérable* de consultation d'information documentaire.

Ce document comprends trois sections principales, l'une pour présenter l'OAI, une autre pour présenter SDX 2 et ses relations possibles avec l'OAI, et enfin une troisième pour évaluer l'intérêt d'effectuer des tests avec le protocole OAI. L'objectif de cette réflexion n'est pas de présenter en détails ce qu'est l'OAI ou SDX, mais bien de voir comment les deux outils peuvent être combinés pour créer des applications documentaires interopérables.

Le protocole OAI

L'OAI est principalement concrétisée par un protocole nommé *The Open Archives Initiative Protocol for Metadata Harvesting*, protocole qui en est à sa version 2³. Ce titre nous renseigne déjà sur deux informations importantes concernant l'OAI :

1. L'OAI s'intéresse d'abord et avant tout aux *métadonnées*, c'est-à-dire à des informations qui décrivent des objets documentaires.
2. Le terme *archives* concerne le nom du projet, mais le protocole lui-même n'a aucune relation avec l'archivistique.

La spécification du protocole présente l'OAI ainsi dans son introduction :

[Le protocole OAI] fournit une infrastructure d'interopérabilité basée sur la récolte de métadonnées, infrastructure indépendante des applications utilisées. Deux classes de participants participent à cette infrastructure : les **fournisseurs de données**, qui vont exposer des métadonnées à propos de certains objets documentaires ; les **fournisseurs de services**, qui vont aller chercher des métadonnées pour offrir des services à certains utilisateurs.⁴

Les fournisseurs de données mettront en place des *entrepôts OAI*, ces entrepôts ayant pour objectif de répondre aux requêtes d'exploration de métadonnées envoyées par des *moissonneurs OAI*, ces moissonneurs étant les outils qui sont mis en place par les fournisseurs de services. Le protocole OAI ne s'intéresse qu'à deux choses dans cette mise en place : comment l'échange entre un moissonneur et un entrepôt s'effectue et comment les métadonnées échangées sont représentées lors de cet échange (et seulement lors de cet échange).

¹ *Open Archives Initiative*, voir <http://www.openarchives.org/>.

² Pour en savoir plus sur SDX, consulter <http://sdx.culture.fr>.

³ La spécification technique complète est disponible à l'adresse <http://www.openarchives.org/OAI/openarchivesprotocol.html>.

⁴ Traduction et accentuation de l'auteur.

Concepts documentaires

Le protocole OAI fait intervenir trois concepts documentaires importants, soit la ressource, l'item et l'enregistrement. Nous allons d'abord présenter ces concepts avant d'aller plus loin dans les explications autour du protocole.

La **ressource** est un objet quelconque, réel ou virtuel, sur lequel on peut définir une description documentaire, des métadonnées, etc. Ainsi, un musée peut être une ressource, de même qu'un livre, un site Web, une base de données, etc. Par exemple, dans un catalogue de bibliothèques, les ressources sont les documents réels que possède la bibliothèque. Le protocole OAI ne s'intéresse pas à ces ressources comme telles ; il suppose qu'elles existent et ont une importance suffisante pour avoir envie de les décrire dans un système documentaire.

L'**item** est l'objet documentaire qui va décrire une ressource ; traditionnellement l'item est une fiche, un enregistrement dans une base documentaire. Cette description peut être quelconque, simple ou complexe, consignée sur un support ou générée à la volée, dans un format ouvert ou propriétaire, etc. Le protocole OAI fait de cet item un *générateur* de métadonnées qui pourront être échangées. La seule restriction imposée par l'OAI pour les items est qu'on puisse y associer un identifiant qui soit unique dans le contexte de l'entrepôt OAI où ils sont stockés et gérés.

L'**enregistrement** est une représentation concrète de métadonnées issues d'un item, c'est-à-dire une représentation dans un format prescrit par le protocole OAI. Ce sont donc des enregistrements qui font l'objet des échanges entre un moissonneur et un entrepôt OAI.

Le protocole OAI laisse donc le champ libre quant au sujet des bases documentaires (quelles sont les *ressources* décrites ?) mais aussi quant à la représentation de ces ressources dans une base sous la forme d'*items*. Le protocole ne s'intéressera qu'à la façon d'échanger des *enregistrements* précis de ces items entre un entrepôt et un moissonneur.

Ainsi, une base documentaire quelconque (par exemple un catalogue de bibliothèque) n'a pas besoin d'être modifiée profondément pour qu'elle soit *compatible* avec l'OAI. Elle peut continuer à fonctionner de la même manière, sauf qu'on doit y ajouter une *couche* permettant d'exposer ses objets documentaires (*items*) selon des formats prescrits par l'OAI, et cela suite à des requêtes précises de la part de moissonneurs OAI, requêtes basées sur le protocole HTTP, donc le Web.

A noter que le protocole prévoit également le concept d'**ensemble**, qui constitue un regroupement quelconque d'items dans un entrepôt OAI. Même si ce concept fait partie intégrante du protocole, un entrepôt OAI n'est pas obligé de regrouper ses items dans des ensembles, et pour la courte présentation que nous faisons ici, la présence ou non d'ensembles ne change rien.

Format d'échange de métadonnées

Les métadonnées documentaires échangées selon le protocole OAI, de mêmes que les informations de gestion, sont nécessairement en format XML. La structure générale de ce format est prescrite par le protocole OAI, elle permet de préciser les éléments de réponse, et en particulier de délimiter les enregistrements qui font partie d'une réponse.

Mais le véritable format qui nous intéresse ici est le format des métadonnées échangées, et le format selon les deux sens du terme : représentation concrète des données (syntaxe) et système descriptif utilisé (sémantique).

La syntaxe des métadonnées échangées doit nécessairement être XML. De plus, le schéma qui définit la syntaxe particulière utilisée dans une réponse doit être référencé explicitement, et doit donc être public. Ainsi, il est tout à fait possible d'utiliser un format comme BiblioML dans une réponse OAI, en autant que l'on puisse mettre en ligne un schéma XML pour le format BiblioML.

La sémantique des métadonnées peut être quelconque. Pour reprendre l'exemple précédent, si on envoie des métadonnées en format BiblioML, la sémantique de ces métadonnées (que signifie l'élément *Title* ?) est définie dans le format BiblioML, et l'OAI ne pose aucune restriction sur les sémantiques utilisées par les formats d'échange de métadonnées.

Le protocole est donc tout à fait générique à ce sujet, mais en même temps, dans le but de rendre les outils plus faciles à réaliser, mais surtout dans le but de rapidement construire des services documentaires basés sur l'OAI, le protocole a ajouté à cette souplesse une obligation pour tous les entrepôts : même si ceux-ci peuvent exposer des métadonnées dans une structure XML quelconque, **ils doivent obligatoirement au moins exposer des métadonnées en format Dublin Core non qualifié**, format dont la sémantique est bien définie¹ et dont la structure XML est définie par un schéma publié par l'OAI.

¹ Voir le site <http://dublincore.org>.

Ce dernier paragraphe explique bien pourquoi on associe généralement le protocole OAI à l'échange de métadonnées en format Dublin Core. D'autres formats (XML) sont possible, mais ce format précis est obligatoire. En d'autres termes, une base documentaire qui veut devenir compatible à l'OAI peut envoyer des métadonnées dans les formats de son choix, mais elle doit pouvoir les envoyer en format Dublin Core, même si en interne elle ne le gère pas dans ce format ; même si en interne son système descriptif est nettement plus riche que le Dublin Core non qualifié.

Le fait d'avoir prévu un format spécifique dans le protocole OAI permet de le garder générique tout en ajoutant une contrainte qui rend la réalisation de bases documentaires à partir de sources différentes *sans même savoir comment sont décrites les ressources originales ni même de quel type de ressources il s'agit*.

Aperçu des actions du protocole

Le protocole OAI prévoit six actions exprimées sous la forme de *verbes*. Ces actions sont celles qui sont initiées par un moissonneur, qui les transmet à un entrepôt, et ce dernier doit réagir en envoyant une réponse au moissonneur.

GetRecord

Cette action consiste à demander un enregistrement précis, normalement à partir de son identifiant. On doit également spécifier le format de données à utiliser ; l'entrepôt est obligé de connaître le Dublin Core seulement. S'il ne connaît pas le format, il peut retourner une erreur.

Identify

Cette action permet d'en savoir plus à propos d'un entrepôt OAI. Un certain nombre d'informations doivent être retournées par l'entrepôt, dans un format précis, comme par exemple un nom, une URL de base, et certaines informations de gestion. L'entrepôt peut également y ajouter d'autres informations, à son gré.

ListIdentifiers

Cette action demande à l'entrepôt de retourner la liste des items qu'il contient, mais seulement avec quelques informations de gestion pour chacun : identifiant, date de dernière modification, ensembles auquel il appartient, etc. Le moissonneur peut demander une liste complète, ou spécifier des critères de date ou d'ensembles.

ListMetadataFormat

Cette action a pour but de connaître la liste des formats de métadonnées disponibles pour un entrepôt.

ListRecords

Cette action est semblable à *ListIdentifiers*, mais cette fois les métadonnées seront également retournées, pas seulement les informations de gestion. Il s'agit donc d'un moyen de récupérer tout ou partie d'un entrepôt OAI.

ListSets

Cette action permet de connaître les ensembles qui ont été définis dans un entrepôt.

Le protocole est donc fort simple. Les échanges se font simplement à l'aide du protocole HTTP, à la base du Web. Les actions et leurs paramètres sont spécifiées par des paramètres ordinaires associés aux échanges HTTP, comme pour tout site Web dynamique. Cette simplicité est l'une des raisons du succès de l'OAI.

SDX 2 et l'OAI

A propos de SDX

La plate-forme SDX est née à la fin de l'année 2000 sous la forme d'une preuve de concept. En effet, l'objectif était de voir quelles pourraient être les fonctionnalités d'un outil de recherche pour bases documentaires XML. Cette preuve de concept ayant été concluante, SDX est devenu un prototype puis une application, la version 1 étant sortie au mois de mai 2001. Un an plus tard, un important chantier a été mis

en place afin de doter SDX d'une architecture plus moderne et plus souple, tout en permettant de répondre à des besoins plus variés. En novembre 2002, la version 2 de la plate-forme voir le jour¹. Depuis les débuts, SDX est livré avec une licence libre².

Fondamentalement, SDX est un outil de recherche pour collections de documents XML. Mais cet outil a été conçu de manière à s'intégrer parfaitement dans une application Web dynamique, voire même de constituer le principal outil de développement d'une telle application. Pour ceci, il s'appuie sur l'architecture Cocoon 2³. Sans aller dans les détails, nous présentons quelques concepts clés de SDX, en particulier ceux qui sont utiles pour une réflexion autour de l'OAI.

Architecture Web

SDX est une application à architecture Web, c'est-à-dire que l'on peut utiliser les applications conçues avec SDX à l'aide d'un navigateur Web, et que SDX lui-même s'installe auprès d'un serveur Web. S'appuyant sur l'architecture Cocoon, SDX utilise l'architecture Web dynamique des *servlets* et est entièrement écrit en Java.

Généricité

SDX est conçu de manière à être totalement générique. En effet, il laisse au développeur d'application le soin de choisir les types de documents à indexer, les champs de recherche à utiliser, comment les remplir, comment afficher les formulaires de recherche, les résultats de recherche, les documents à consulter, etc.

Indexation et recherche

Une base de documents SDX est un lieu où s'effectuent des recherches documentaires. Cette base est constituée de *champs de recherche* ; la recherche s'effectue toujours dans un ou plusieurs champs. La relation entre les structures XML indexées et les champs définis dans une base est sous l'entière responsabilité du développeur de l'application.

Structure des applications

Une *application SDX* est un site Web dynamique où un certain nombre de services sont offerts aux utilisateurs. Une application comprendra normalement une ou plusieurs *bases de documents*, qui, tel qu'évoqué, sont les liens où s'effectuent des recherches. Les documents indexés par SDX doivent être par la suite accessibles pour la consultation, et pour les rendre accessibles SDX les stocke dans des *entrepôts*. Ces entrepôts peuvent être sous le contrôle de SDX ou non, et dans ce second scénario SDX ne conservera que l'adresse des documents.

SDX, entrepôt OAI ?

SDX, en tant qu'outil de recherche pour documents XML, stocke des objets documentaires que sont ces documents XML. La recherche dans SDX étant basé sur la notion de *champ*, et une base de documents SDX étant une collection *plate* de champs, on peut également considéré que SDX stocke et gère des métadonnées à propos des objets documentaires indexés, ces métadonnées étant le contenu des champs définis pour une base de documents. Autrement dit, SDX est déjà un *serveur de métadonnées relatives à des objets documentaires*.

Il est donc tout à fait naturel de considérer SDX 2 comme un excellent candidat pour servir d'entrepôt OAI. Ou plus précisément, une base de documents d'une application SDX 2 pourrait être un entrepôt OAI, SDX servant ici de relais qui écoute les requêtes OAI et qui répond en fonction du contenu d'une base. Ce n'est donc pas une installation SDX qui serait un entrepôt, cette installation pourrait être une interface pour un grand nombre d'entrepôts. Pour ce faire, il manque essentiellement deux aspects que nous évoquons ici.

Réponse à des requêtes OAI

Un entrepôt OAI doit être en mesure de répondre à des requêtes utilisant le protocole OAI, requêtes qui proviennent d'une connexion HTTP. SDX évolue déjà sur un tel protocole, alors il serait tout à fait

1 Au moment d'écrire ces lignes, SDX 2.0 est en version beta ; la version définitive devrait être disponible avant la fin du mois de novembre 2002.

2 Plus précisément, la licence est la GPL de la Free Software Foundation, voir <http://www.gnu.org/licenses/gpl.html>.

3 Voir <http://xml.apache.org/cocoon/>.

envisageable d'ajouter une couche OAI lui permettant de fournir des métadonnées à des moissonneurs distants qui voudraient en savoir plus sur les ressources documentaires disponibles.

Formats de données

Les métadonnées dans SDX sont les champs faisant partie des résultats de recherche ; ces informations sont dans une structure XML simple mais propre à SDX. On peut aussi considérer, dans certains cas, que les documents XML indexés sont eux-mêmes des métadonnées, donc toujours en format XML mais dont la structure peut être quelconque.

Des formats XML divers peuvent être échangés dans une transaction OAI, mais rien n'assure qu'elles seront correctement gérées aux deux extrémités. C'est pourquoi il est nécessaire que SDX puisse envoyer des données en format Dublin Core. De façon générale, SDX ne possède pas d'information Dublin Core à propos des documents qu'il gère, alors il faudrait ajouter une façon d'indiquer cette information.

La description des champs d'une base de documents se fait à l'aide d'un fichier de configuration en format XML. Pour chaque champ, des informations telles que la langue, le type d'indexation, le nom du champ se trouvent dans cette description. Il serait très simple pour un développeur d'application d'indiquer quel est l'équivalent Dublin Core des champs pertinents dans une base de documents, et ensuite SDX se chargerait d'envoyer une structure Dublin Core adéquate pour chaque objet indexé lorsqu'on lui en fait la demande.

SDX 2 n'est pas actuellement programmé pour être un entrepôt OAI, mais sa structure interne le rend tout à fait apte à le devenir. Une fois le protocole OAI implanté au niveau de SDX, un effort minimal de la part des développeurs SDX (indiquer des équivalences entre leurs champs et les champs Dublin Core) permettrait de rendre toute application SDX existante immédiatement accessible via le protocole OAI.

On pourrait également prévoir une structure de définition d'ensembles OAI dans une base de documents SDX. Les ensembles pourraient, par exemple, être définis par des requêtes de recherche SDX, mais on pourrait également prévoir de fournir des informations sur l'appartenance à un ensemble lors de l'indexation du document.

SDX, moissonneur OAI ?

Le moissonneur OAI a pour unique d'interroger des entrepôts OAI dans le but de collecter des métadonnées à propos d'objets documentaires. Ce qu'il fait de ces métadonnées n'est pas prévu dans le protocole, mais il est certain que la recherche dans ces métadonnées et l'affichage des métadonnées ou des objets documentaires qu'elles décrivent constituent deux fonctions que l'on cherchera à implanter dans un projet OAI. SDX étant à la fois un outil de recherche et un outil de consultation d'objets documentaires, lui faire jouer également un rôle de moissonneur OAI semble tout à fait logique et envisageable.

Encore une fois, il manque deux aspects à SDX pour qu'il puisse prétendre être moissonneur SDX, et nous les expliquons ici.

Exprimer des requêtes OAI

Le moissonneur est l'outil qui va initier une transaction par l'envoi d'une requête OAI. Cette requête utilise un protocole particulier qu'il est nécessaire d'implanter. De plus, il faudrait prévoir dans la configuration d'une base de documents SDX intéressée par les fonctions de moissonneur, un moyen de préciser la fréquence de moisson souhaitée, ou encore être capable de déclencher une telle moisson de façon applicative.

Stocker et indexer les données

Suite à l'envoi d'une requête OAI, un moissonneur récoltera habituellement des métadonnées, le plus souvent en format Dublin Core. Pour être utiles, ces données devraient être indexées dans une base de documents SDX.

On peut entrevoir cette indexation de trois façons. Soit il s'agit de métadonnées Dublin Core que l'on doit insérer dans une base de documents spécifiquement dédiée à cette fin, et dans un tel cas SDX peut prendre en charge totalement cette indexation. Soit il s'agit de métadonnées en format Dublin Core ou d'autres formats, mais qui sont à insérer dans une base de documents qui sert à d'autres fins, dans un tel cas SDX pourrait envoyer les métadonnées dans le processus d'indexation normal de cette base de documents, processus qui prendrait donc en charge cette indexation.

Ce deuxième aspect illustre bien le caractère générique de SDX. Les données issues d'un entrepôt OAI sont en XML, alors il n'y a aucune difficulté à les indexer dans une base de documents. Par contre, SDX n'a rien

prévu pour l'indexation de structures précises telles que le Dublin Core, alors il est nécessaire d'implanter cet aspect spécifique. On pourrait même aller plus loin et prévoir, comme complément à SDX, toutes les fonctionnalités nécessaires pour réaliser une application documentaire complète (recherche, consultation, alimentation via OAI ou des formulaires) basée sur Dublin Core.

Une chose est certaine, encore une fois l'architecture et les fonctionnalités SDX sont totalement en phase avec les fonctions nécessaires pour implanter un moissonneur OAI, alors il est tout à fait envisageable de faire de SDX un tel moissonneur.

Pourquoi l'OAI avec SDX ?

Il existe déjà des outils qui permettent d'implanter les fonctionnalités d'entrepôt et de moissonneur OAI, alors pourquoi chercher à les implanter dans SDX ? Tout d'abord, il est bon de souligner que le fait d'avoir plusieurs outils pour remplir une fonction n'est pas nécessairement une mauvaise chose, y compris dans un univers de logiciels libres, car cela permet de créer une saine compétition entre différents projets et de faire avancer les développements.

Mais surtout, SDX est utilisé pour plusieurs applications où il serait intéressant de pouvoir échanger des métadonnées avec le protocole OAI, alors plutôt que de se tourner vers des outils complémentaires, il est intéressant d'envisager un support OAI dans SDX.

Tester l'OAI

On pourrait imaginer plusieurs types de tests pour vérifier la conformité des outils OAI d'une part, mais surtout l'intérêt de l'échange de métadonnées avec ce protocole dans différents contextes. Nous allons toutefois exclure des tests à vocation principalement technologiques pour mettre l'accent sur des tests qui s'intéressent principalement aux données et à leur échange.

Données homogènes réparties

Il arrive fréquemment que des données tout à fait homogènes soient réparties sur plusieurs serveurs. Prenons pour exemple des données issues des différents services de l'inventaire associés au ministère de la culture. Ces données sont en général accumulées par les services régionaux de l'inventaire selon une méthodologie détaillée qu'ils respectent tous, et on pourrait penser qu'elles seraient diffusées sur des serveurs régionaux. Par ailleurs, il serait naturel de prévoir un service où les utilisateurs pourraient interroger ces données sans nécessairement passer par un serveur régional, par exemple parce que les besoins de recherche concernent plusieurs régions.

Le protocole OAI peut être une solution, et ce cas de figure est peut-être le plus simple à mettre en place, car ici les structures de données et les vocabulaires sont les mêmes, ce qui facilite les échanges, ou plus spécifiquement les fonctionnalités attendues suite aux échanges. Mais un tel test permettrait de vérifier certains aspects comme les fréquences optimales de mise à jour, l'impact de la recherche sur les métadonnées plutôt que sur les objets documentaires eux-mêmes, etc.

Structures homogènes réparties

Ce type de test est semblable au précédent, dans le sens où les données sont réparties et sont toutes structurées de la même façon. Toutefois, la façon d'utiliser ces structures peuvent varier, notamment au niveau des vocabulaires.

Imaginons par exemple un certain nombre de services de documentation qui constituent des catalogues bibliographiques. Ces services, dans un souci d'harmonisation, pourraient choisir d'utiliser un format commun, par exemple le format BiblioML¹. Toutefois, ces services n'utilisent pas tous BiblioML de la même façon ; certains feront des notices très élaborées, d'autres plutôt simples. Certains effectueront une indexation géographique, d'autres non. Certains feront une indexation sujet avec un thésaurus particulier, d'autres avec un vocabulaire libre ou un autre thésaurus.

¹ Le format BiblioML permet de représenter des données bibliographiques simples ou complexes en format XML. Voir <http://www.biblioml.org>.

L'utilisation de l'OAI pour réaliser une base de recherche commune est possible dans un tel contexte, mais la cohérence de l'ensemble sera moins grande. Il serait toutefois très intéressant d'effectuer ce genre de tests justement pour évaluer cette cohérence potentielle et quels sont les facteurs qui permettent de l'augmenter.

Données hétérogènes

De plus en plus de sites Web jouent un rôle de *portail d'accès à l'information*, information qui bien souvent est de nature fort diverse (pages HTML, bases de données, documents XML, etc.) et parfois répartie, mais pas toujours. Dans un tel scénario, l'OAI peut aussi être intéressant car il offre un moyen de récupérer un minimum de données compatibles parmi toutes les structures existantes ou même l'absence de structure.

Il serait donc très intéressant de faire un test dans un contexte de données très hétérogènes et plus ou moins structurées, afin de mesurer l'impact organisationnel et documentaire de l'utilisation d'un protocole d'échange.

Conclusion

Le protocole OAI s'impose aujourd'hui comme l'une des principales méthodes de mise en commun de ressources documentaires plus ou moins hétérogènes et plus ou moins réparties. Dans le contexte du ministère de la culture et de la communication, plusieurs services ont mis en place des bases documentaires à l'aide de la plate-forme SDX, et nous avons montré qu'il est tout à fait réaliste d'ajouter un support OAI à SDX 2. C'est pourquoi il pourrait être intéressant d'effectuer des tests pour vérifier le véritable intérêt de ce protocole, à l'aide d'outil, de données et de contextes réalistes.

Toutefois, dans l'état actuel des technologies, il est nécessaire de considérer ces tests comme de véritables applications qui répondraient à des besoins précis sur des données existantes, ce qui aurait en plus une valeur de démonstration très forte. Pour y arriver, il faut non seulement trouver des aspects à tester, mais également des données susceptibles de faire partie de ces tests, et des institutions susceptibles de participer à ces tests, soit au développement soit à l'évaluation des résultats.