Date : 28/07/2006

Un point sur les nouveaux protocoles de recherche d'information : SRU, OpenSearch/A9, CQL et Xquery

Sally H. McCallum
Library of Congress
USA

Traduction : Sylvie DALBIN, Assistance & Techniques Documentaires (22 juillet 2006)

Meeting:	**102 IFLA-CDNL Alliance for Bibliographic Standards ICABS**
Simultaneous Interpretation:	Yes

WORLD LIBRARY AND INFORMATION CONGRESS: 72ND IFLA GENERAL CONFERENCE AND COUNCIL
20-24 August 2006, Seoul, Korea
http://www.ifla.org/IV/ifla72/index.htm

Résumé

Les bibliothèques sont largement concernées par les protocoles de recherche d'information : en effet, les systèmes de gestion de bibliothèques sont variés alors que les lecteurs veulent accéder à différents sites sans avoir à connaître la syntaxe d'interrogation propre à chacun. Cet article présente plusieurs protocoles de recherche et langages d'interrogation parmi les plus récents, et compare leurs avantages : Search via URL (SRU), OpenSearch Contextual Query Language (CQL) et XQuery. Les présentations des services de SRU et d'OpenSearch expliquent leurs différences fonctionnelles – recherche par mots clés et fourniture des réponses en format unique pour OpenSearch, recherche plus riche avec des formats de réponse multiples pour SRU. Les avantages de CQL sont décrits avec leur possibilité d'utilisations complémentaires du langage très fouillé et complexe, XQuery, développé pour XML.

Introduction

La recherche d'information, qui semble si simple et si évidente avec Google, est étonnamment complexe. Jusqu'au développement de l'informatique vers la moitié du siècle précédent, la recherche de documents dépendait essentiellement des métadonnées – alors appelé catalogage – qui étaient placées dans des dispositifs sous forme de fiches ou de catalogues papier que devaient compulser les utilisateurs.

Cette « technologie » a donné lieu au développement et à la fine instauration d'un nombre important de conventions visant à améliorer la recherche – jeux de métadonnées compatibles entre eux, règles pour la formulation des points d'accès, listes de noms, conventions pour les renvois de termes associés, développement extensif des mots matière généraux ou spécialisés, développement de chaînes de mots matière dont l'ordre présente un sens en lui-même. Toutes ces composantes ont donné lieu à des métadonnées très contrôlées, rassemblées dans les fichiers pour aider les lecteurs dans leurs recherches… et qui se sont révélées être des tests de mémoire ou d'association d'idées très performants !

Avec l'émergence des ordinateurs, les spécialistes de l'information ont commencé à utiliser les métadonnées d'une nouvelle manière pour améliorer les recherches des utilisateurs en exploitant les relations entre les métadonnées. Dans les années 1980, les bibliothèques et les fournisseurs de systèmes de bibliothèques ont été très créatifs en utilisant les métadonnées normalisées au format MARC pour proposer aux utilisateurs une grande variété de méthodes de recherche – et de produits. Ces systèmes de gestion de bibliothèques ayant des architectures différentes, les bibliothèques ont développé à la fin des années 1980, un protocole de recherche normalisé offrant des possibilités d'interrogation entre systèmes ; ce protocole s'est déployé dans les années 1990. Ceci a été rendu possible grâce à la forte normalisation des métadonnées au sein de la communauté : le protocole s'est concentré sur les différences entre les systèmes – en particulier sur les architectures de base et les approches conceptuelles –, qui correspondaient à la réalité pré-historique du Web. Aujourd'hui ce protocole, Z39.50 ([i]) est devenu une composante essentielle des catalogues de bibliothèques. Par exemple, l'interface Web du catalogue de la Bibliothèque du Congrès reçoit une moyenne approximative de 150 000 recherches par jour, dont plus de la moitié provient du protocole Z39.50.

Le développement et l'utilisation considérables du protocole Z39.50 ont mis en évidence la place centrale que joue la normalisation de l'indexation dans l'interrogation de catalogues à distance. La disparité entre indexations causait en effet une grande confusion chez les utilisateurs. Des efforts ont été faits à plusieurs reprises pour produire des « index standardisés », mais il n'y a jamais eu de réel enthousiasme de la part des fournisseurs de systèmes, parfois même de la réticence. Certes, la standardisation des métadonnées permet de les partager, mais on peut aussi souligner que la liberté d'indexation du format MARC a été bénéfique aux bibliothèques en encourageant les développeurs de systèmes à expérimenter différentes approches de recherche documentaire.

Si nous faisons défiler ces 20 dernières années, jusqu'à 2006, nous remarquons que les technologies ont offert beaucoup de possibilités pour l'interrogation multi-plateformes. Le Web a été inventé dans les années 1990, et l'essor de XML a fourni une occasion d'adapter à Internet les mécanismes de recherche précédemment développés pour Z39.50. Depuis 2000 on a vu naître nombre de solutions tirant profit de ces nouveaux outils, de nouvelles normes et de la profusion de documents numériques disponibles en texte intégral. Dès lors, les composantes de la recherche documentaire – métadonnées, vocabulaires contrôlés, associations, mots clés libres, indexation, rappel et pertinence – ont été étudiées sous un éclairage nouveau.

La recherche documentaire est un vaste sujet, aussi cet article se limite à examiner quelques modèles et protocoles actuels développés pour la recherche multi-bases. En particulier, nous nous concentrerons sur deux protocoles de recherche, Search/Retrieve via URL (SRU) et OpenSearch/A9, et deux langages de requête, Contextual Query Language (CQL) et Xquery. La méta-recherche ou

recherche multisites n'est pas traitée en tant que telle ici, étant donné que pour la plupart des protocoles client/serveur, elle est configurable pour fournir à l'utilisateur final les résultats d'une méta-recherche – avec plus ou moins de sophistication et de transparence.

Terminologie utilisée

Cet article différencie un *langage de requête* d'un *protocole de recherche*. Un protocole de recherche correspond à une série de messages entre un client (appelé *utilisateur* dans cet article) et un serveur (appelé *cible*). L'utilisateur n'est donc pas ici l'utilisateur final, mais le système qu'il utilise. Le requête proprement dite est un élément de ces messages; qui contiennent aussi des informations sur le contexte de la requête, les préférences de l'utilisateur ou de la cible, le mode de traitement pour répondre à la requête, et le moyen d'acheminer les résultats récupérés (la *récupération* des réponses). Le *protocole* est généralement utilisé pour gérer les demandes de recherche et récupérer les données, alors que la r*equête* sert à formuler les critères de recherche. Le *langage de requête* fournit la syntaxe qui précise, dans une forme structurée, les paramètres de la requête telle que « rechercher le terme exact 'poisson' dans les titres ou les sujets ». Une distinction importante doit être faite entre une requête dans une syntaxe locale ou utilisateur, et une requête utilisant une définition abrégée ou normalisée de cette syntaxe.

SRU – Search/Retrieval via URL

SRU est un protocole de recherche et de récupération d'information qui utilise les structures Internet et Web pour transporter les messages entre utilisateurs et cibles. SRU a un précédent important : Z39.50, protocole très largement mis en œuvre. La plupart des fonctionnalités de SRU dérivent de cet ancien protocole, mais seules les plus utiles ont été conservées, sous une forme simplifiée. Tandis que le développement de SRU a commencé, des recherches similaires utilisant les URL sont à l'étude dans plusieurs institutions, notamment à la Bibliothèque Royale des Pays-Bas. Un groupe international d'experts de la communauté Z39.50 collabore sur un projet concernant ce nouveau protocole dans l'environnement Internet/Web/XML. Les spécifications SRU ont été publiées pour la première fois en 2002, et se sont popularisées dans des applications nouvelles en raison de leur facilité d'implémentation ([ii]).

SRU est très flexible. Il est basé sur XML. L'implémentation la plus courante est SRU via URL qui utilise la commande GET HTTP pour le transfert des messages. Mais d'autres versions peuvent aussi fonctionner avec le protocole SOAP (SRU via SOAP) qui peut accueillir davantage de fonctionnalités web, et la commande POST HTTP (SRU via POST) qui évite certaines restrictions sur la longueur ou sur les jeux de caractères, restrictions qui existent dans GET HTTP. Les données peuvent être fournies en réponse à la demande dans n'importe quel format XML bien défini.

Le modèle de Z39.50 et de SRU

Le modèle fonctionnel pour la recherche multisites avec le protocole SRU est identique au modèle utilisé pour le protocole Z39.50. L'utilisateur final lance une requête sur le système utilisateur (source), qui utilise la syntaxe spécifique locale. Pour effectuer une recherche dans un système cible avec sa syntaxe de recherche spécifique, sa propre structure de base de données et ses règles d'indexation, la requête locale de l'utilisateur est transformée dans un format standard. La cible reçoit ce message de recherche standard, composé d'une requête et d'un protocole d'échange, et le traduit dans une syntaxe compréhensible par ses propres bases de données. Si la requête est très détaillée ou que les paramètres de la recherche ne peuvent pas être pris en compte par la cible, celle-ci rejette la requête ou, dans d'autres cas, ignore certains attributs qu'elle ne peut traiter et effectue la recherche en mode dégradé.

Le protocole de recherche qui accompagne la requête précise à la cible les préférences de l'utilisateur : le format des enregistrements à récupérer, leur nombre, d'autres caractéristiques concernant l'éventail des réponses, etc. Le protocole charge alors les réponses, qui peuvent inclure des notices récupérées, des indications d'erreurs, des spécifications sur le format des notices récupérées, etc.

Puisqu'un des points essentiels d'une recherche inter-systèmes porte sur les différences d'indexation d'un site à l'autre, les développeurs Z3950 et SRU ont amélioré la qualité de recherche inter-systèmes en créant une fonctionnalité qui permet à l'utilisateur de demander en premier lieu à la cible sous quel système elle fonctionne, en espérant que la cible donne ces indications selon les normes spécifiées dans le protocole. En utilisant l'information obtenue concernant les index des cibles, on espère ainsi que le système utilisateur formulera une requête plus aboutie.

Les services SRU

SRU propose trois services de base : *Explain* (expliquer), *Search/Retrieve* (rechercher/récupérer) et *Scan* (balayer).

SRU Explain. S'appuyant sur l'expérience considérable acquise avec le protocole Z39.50, les développeurs de SRU ont traité en priorité le service Explain, qui permet à la cible d'indiquer à l'utilisateur ce qu'elle lui permet de faire. La cible se décrit elle-même dans un format standardisé XML, facilement récupérable par le système utilisateur via une requête Explain. Cette description peut aider l'utilisateur à formuler sa requête. Explain a de nombreux autres champs descriptifs, comme les formats XML sous lesquels peuvent être fournis les résultats, ou le nombre de réponses fournies par défaut. Les cibles SRU ne sont pas obligées d'avoir un document Explain, mais elles y sont fortement incitées.

SRU Search/Retrieve. L'opération Search/Retrieve est la fonction principale de SRU. Elle gère l'envoi de la requête à la cible selon les préférences indiquées par le protocole, et gère la réponse à cette requête, c'est-à-dire l'envoi des données et des informations associées. La syntaxe de la requête utilisée est le *Contextual Query Language* décrit ci-dessous.

SRU Scan. SRU Scan permet à l'utilisateur de consulter les termes proposés par une cible autour d'un terme spécifié dans les paramètres, ainsi que le nombre d'occurrences de chacun de ces termes — si la cible propose la fonctionnalité de balayage. Alors que Search/Retrieve permet de rechercher des réponses dans des index de termes, Scan permet à l'utilisateur de demander à récupérer un ensemble de termes à un point précis de cet index. Il peut ainsi visualiser une liste ordonnée de termes et, si cette fonctionnalité est proposée par le système, connaître leur nombre d'occurrences. Scan est souvent utilisé pour sélectionner des termes avant de lancer une recherche, ou pour vérifier visuellement les résultats négatifs d'une recherche.

Exemple d'utilisation. SRU a été implémenté initialement par des sites Z39.50 pour offrir une possibilité supplémentaire d'accès aux catalogues, ainsi que par des sites non Z39.50 pour faciliter les recherches sur des ressources externes. La Bibliothèque du Congrès a implémenté SRU comme serveur cible en 2004, en utilisant une passerelle vers l'interface Z39.50 proposée dans le catalogue du fournisseur du logiciel au lieu d'offrir un accès direct aux bases de données. Ce logiciel, qui incorpore le protocole SRU, a aussi permis à la Bibliothèque du Congrès de tester les réponses directes Z39.50 et de fournir des réponses aux format MARC 21, MARCXML, MODS et même en DC (Dublin Core). La Bibliothèque du Congrès est en train d'implémenter SRU pour des bases de données qui ne sont pas natives en MARC, mais il est encore tôt pour penser à fournir des réponses en MARCXML ou MODS.

CQL – Contextual Query Language (langage de requête contextuel)

Un éléments-clé du service Search/Retrieval est la requête. Les concepteurs de SRU ont développé une syntaxe de requête à la fois riche et simple — ils ont su ajuster finement le protocole pour obtenir le meilleur des métadonnées des bibliothèques. Ce langage de requête est le Contextual Query Language (ou langage de requête contextuel), nommé couramment CQL ([iii]).

CQL est un langage formel de présentation des requêtes. Il a été conçu pour s'adapter aux systèmes de recherche d'information tels que les index Web, les catalogues bibliographiques ou les informations de

collections de musées. A la différence de la syntaxe de requête généralement utilisée avec Z39.50, CQL cherche à produire une lecture et une écriture humaines, à s'apparenter à un raisonnement intuitif, tout en conservant des composantes importantes du langage de requête Z39.50, très expressif et très complexe. C'est pourquoi ce langage est plus puissant qu'un simple langage de requête de type Google. On lit sur le site Web du SRU que « CQL essaie de combiner simplicité et expressivité intuitive pour des requêtes simples et fréquentes, tout en offrant la richesse des langages les plus expressifs pour exprimer, si nécessaire, des concepts plus complexes »([iv]). Bien que CQL cherche à être lisible par l'homme, il est toutefois prévu que les utilisateurs finaux puissent l'exploiter à travers une interface conviviale qui soit simplement l'interface et la syntaxe de leur catalogue local.

CQL est fondé sur la définition d'un ensemble de points d'accès abstraits tels que le titre, l'auteur, le sujet, et leurs déclinaison, par exemple les auteurs physiques, les titres uniformes et les sujets géographiques. Alors que les grandes bases de données possèdent généralement une forme d'index et que les points d'accès « abstraits » du CSL soient souvent appelés « index » abstraits, CQL n'impose aux serveurs cibles d'avoir des index « physiques », mais il exige la possibilité d'obtenir des réponses comme si de tels index existaient. CQL ne fait pas d'hypothèse sur le modèle de la base de données — si celle-ci est relationnelle, objet, hiérarchique, réseau, etc. —, mais influence la recherche : il vise des métadonnées identifiées (les notices sont considérés comme des données plutôt que comme des documents) afin d'effectuer une recherche « intelligente ». En effectuant des recherches simples, ce service ne déploie pas son potentiel.

Un modèle différent, OpenSearch

En mars 2005, Amazon a sorti un nouveau service appelé OpenSearch, une implémentation du protocole de recherche développé par l'éditeur de logiciel A9 ([v]). Alors qu'OpenSearch/A9 peut être associé à un certain nombre de services comme les fils RSS, nous nous bornerons ici à examiner le protocole de recherche d'OpenSearch : son modèle fonctionnel, ses différences avec SRU, les avantages comparatifs et combinaisons possibles entre les deux approches.

Comme les bibliothèques, Amazon reconnaît que « les moteurs de recherche traditionnels ne parviennent souvent pas à indexer correctement le contenu de sites Web spécialisés, et que les moteurs de recherche locaux peuvent mieux traiter les contenus locaux : 'Des contenus différents appellent des moteurs de recherche différents. La plupart du temps, le meilleur moteur de recherche pour un site est celui qui est créé par ceux qui connaissent le mieux le contenu de ce site' »([vi]). Il est évident, par exemple, qu'un catalogue dans le domaine médical permet de lancer des recherches et d'obtenir des réponses très spécialisées et précises. Même dans des bases de données généralistes, la recherche peut être profilée pour un public spécifique, par exemple pour des élèves, des chercheurs ou des étudiants en art. Dès lors il peut y avoir une diversité fonctionnelle dans les techniques de recherche documentaire. Les protocoles de recherche externes comme Z39.50 et SRU ont bien identifiée la question de la diversité d'indexation , pour laquelle ils cherchent à fournir des solutions.

Dans leur volonté de développer des modes spécialisés de recherche d'information, Amazon et A9 ont développé le modèle suivant : le système utilisateur demande à la cible des informations sur son système. La cible renvoie les noms des paramètres locaux des tâches de recherche. L'utilisateur envoie alors une requête à la cible en utilisant le "langage" de la cible et reçoit les notices récupérées par la cible au format RSS. Le concept de base est intéressant; les interactions sont le plus souvent compatibles avec SRU tout en évitant tant que possible la complexité. OpenSearch facilite les recherches simples par mot-clés, mais par contre est sensiblement plus délicat à mettre en oeuvre pour des recherches plus précises.

Le client OpenSearch demande en premier des informations minimales à la cible — en particulier les noms des paramètres employés par la cible qu'il utilisera dans le formulaire de recherche locale. Les sous-champs des points d'accès de la cible ne sont pas renseignés. Il n'est pas nécessaire dans une syntaxe standard de recherche comme CQL de lancer des requêtes fines, car ces requêtes sont uniquement constituées de mots clés. Il n'y a aucune possibilité de choix de syntaxe pour les réponses.

OpenSearch est un protocole utile pour les recherches très simples, essentiellement pour des requêtes par mot-clés. Devant le fait que différentes bases de données puissent avoir des modalités de recherche différentes, il envoi simplement un mot-clé que la cible traitera de la manière qui lui sera la plus appropriée. On suppose que l'utilisateur final n'a pas besoin de connaître la façon dont le terme a été traité par la cible.

OpenSearch et SRU présentent des différences. OpenSearch ne cherche pas à comprendre la cible mais propose simplement à la cible de traiter une recherche de termes limités à des mots-clés. SRU peut traiter des recherches simples similaires, mais il a également la capacité d'accéder à une compréhension de la cible qui lui permet d'interpréter les spécificités des requêtes utilisateurs et donc de fournir des résultats plus fins, en particulier grâce au service Explain et au développement de points d'accès abstraits. SRU traite également des requêtes avec une syntaxe spécifique en XML pour les réponses, tandis qu'OpenSearch a simplifié son protocole en restreignant le format des réponses à une syntaxe RSS.

L'approche d'OpenSearch est particulièrement intéressante pour des recherches sur des serveurs et des fichiers non structurés. Bien que les résultats puissent être de qualité variable, ils sont suffisants pour remplir certains objectifs, ou pour servir de première sélection de sites sur lesquels effectuer ensuite des recherches locales. SRU est mieux utilisé dans des catalogues structurés, lorsque l'utilisateur souhaite conduire une recherche plus structurée et mieux vérifier les réponses fournies. Les utilisateurs habitués à un catalogue local dans une bibliothèque trouveront SRU plus pertinent en raison de sa proximité avec les outils de recherche locaux ; des utilisateurs habitués à une recherche sur Google, trouveront les résultats OpenSearch plus satisfaisants.

Les deux protocoles sont compatibles et des discussions ont été initiées entre les développeurs de A9 et ceux de SRU([vii]). L'objectif serait de permettre à OpenSearch de mener des recherches plus riches, et relier des environnements qui n'exigent pas un protocole strict comme SRU, mais qui souhaitent offrir plus de fonctionnalités que n'en fournit actuellement OpenSearch.

XQuery

XQuery, développé par le World Wide Web Consortium (W3C), est un langage de requête pour des données XML ([viii]). Les travaux progressent lentement depuis environ 7 ans, avec un désaccord fondamental entre ceux qui voient XML comme un langage de balisage de documents (identifiant des paragraphes, débuts de chapitres…) et ceux qui voient XML plutôt comme un langage de balisage de données (identifiant des noms, sujets, dates…). XQuery a besoin de connaître la sémantique de balisage XML des documents recherchés. Il davantage orienté vers l'exploitation des documents primaires que des index. Mais XQuery, langage de requête complexe (et complet, si quelque chose dans ce secteur puisse être considéré comme complet), peut trouver sa justification principale dans le traitement des requêtes à l'intérieur d'un système, plutôt qu'entre des applications clients/serveurs. SRU et OpenSearch, qui sont des protocoles client/serveur, peuvent interagir avec des applications XQuery quand ils accèdent à un tel serveur comme c'est le cas avec SQL (Structured Query Language) ou d'autres logiciels de recherche de bases de données.

En conclusion

Cet article a permis d'explorer quelques-uns des « acronymes » les plus populaires dans des environnements de recherche client/serveur : des protocoles de recherche comme SRU et OpenSearch/A9, et des langages de requêtes comme CQL ou XQuery.

Les bibliothèques ont besoin de mettre en œuvre des approches variées, des recherches « intelligentes » de données structurées dans des catalogues et non-structurées dans des documents numériques et des répertoires. SRU et OpenSearch constituent des solutions innovantes, étudiées pour répondre à la diversité des systèmes de recherche des serveurs cibles. La collaboration entre développeurs est une bonne nouvelle, et devrait conduire, nous l'espérons, à des améliorations de ces

deux approches. En attendant, CQL offre une syntaxe de recherche contrôlée et modérément structurée pour des applications client/serveur — aujourd'hui adaptée à une variété de serveurs particuliers ; tandis que XQuery, complet et très complexe, peut fournir une base solide pour de futurs développements pour la recherche.

---

[i] Le site web du protocole Z39.50 est le : <http://www.loc.gov/z3950/agency/>. S'y trouvent des liens vers la version ANSI/NISO de la norme, et une copie de la version ISO de la norme fournie par l'ISO (ISO23950)

[ii] Le site Web SRU est le : <http://www.loc.gov/standards/sru/>. Les spécifications, descriptions des développements ainsi que la documentation SRU sont disponibles sur ce site.

[iii] Le site Web CQL est le : <http://www.loc.gov/standards/sru/cql/>

[iv] <http://www.loc.gov/standards/sru/cql/>. (Consulté le 30 Mai 2006)

[v] Wiggins, Richard W., Amazon's New OpenSearch Enables Search Syndication©, 28 Mars 2005 <www.infotoday.com/> (Consulté le 22 avril 2006)

[vi] La FAQ OpenSearch est localisée à : <http://opensearch.a9.com/docs/faq.jsp> (Consulté le 22 Avril 2006)

[vii] « Visite de Rob Sanderson à A9.com » <http://blog.a9.com/blog/2006/04/14/rob-sanderson-visits-a9com/> (Consulté le 29 Juin 2006)

[viiiviii] XML Query (XQuery) Requirements©, W3C Working Draft 3 June 2005 <http://www.w3.org/TR/xquery-requirements> (Consulté le 30 Mai 2006)