



Date : 14/08/2006

Ein Blick auf neue Protokolle zum Information Retrieval: SRU, OpenSearch/AQ9, CGL, and Xquery

Sally McCallum ¹
Library of Congress
USA

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>

Abstract

Bibliotheken haben großes Interesse an Suchprotokollen, weil Bibliothekssysteme verschieden sind, und Bibliotheksbenutzer dennoch den Zugang zu verschiedenen Sites benötigen, ohne die Suchsyntax jeder Seite zu lernen. Dieser Vortrag untersucht und vergleicht die relativen Vorteile von einigen der aktuellsten Suchprotokollen und Retrievalsprachen: Search via URL (SRU), OpenSearch, Contextual Query Language (CQL) und XQuery. Die Modelle für den SRU- und OpenSearch-Einsatz werden beschrieben, um die Unterschiede bei der Funktionalität zu erläutern – Stichwort-Suche mit einfacher Wiedergabe von Datensätzen bei OpenSearch und umfassendere Suche mit mehrfacher Wiedergabe von Formatdaten bei SRU. Die Vorzüge von CQL werden ebenso beschrieben wie möglicherweise ergänzende Anwendungen des sehr detaillierten und komplexen XQuery, das für XML entwickelt wurde.

Einleitung

Das Information Retrieval, das einfach und klar erscheint, wenn man Zugang zu Google hat, ist überraschend komplex. Bis zu dem Zeitpunkt, zu dem die elektronische Datenverarbeitung während des letzten Jahrhunderts halbwegs entwickelt war, hing das Auffinden von Dokumenten in hohem Maße von Metadaten ab – damals Katalogisierung genannt – die auf Vorrichtungen wie Karten- oder Bandkatalogen installiert/angebracht werden konnten und dem Suchenden zum Durchsehen angeboten wurden. Diese „Technologie“ führte zur Entwicklung und Verfeinerung einer Zahl von Vereinbarungen zur Verbesserung des Retrieval – abgestimmte Sets von Metadaten, die erfasst werden sollen, von Regeln zur Bestimmung von Einstiegspunkten, Namenslisten, Vereinbarungen zur Cross-Recherche, um die Zuordnungen von gleichen Namen sicherzustellen, umfassende Entwicklung von allgemeinen und speziellen Fachvokabularen und Entwicklung von Schlagwortketten, deren Bedeutung durch ihre Reihenfolge festgelegt ist. Alle diese Bestandteile haben zu hochgradig standardisierten Metadaten geführt, die in Dateien zusammengestellt wurden, die man durchsuchen konnte, und die bei Tests zum Wiederauffinden und zur Relevanz gut abgeschnitten haben.

Mit der Ankunft von Computern haben Informationsspezialisten angefangen, für ein verbessertes Benutzerretrieval die Metadaten auf neue Arten zu verwenden, indem sie die Konsistenz der Metadaten ausnutzten. In den Achtziger Jahren des letzten Jahrhunderts waren Bibliotheken und Bibliothekssystemverkäufer ziemlich kreativ. Sie haben die Standard-Metadaten in einer MARC-Aufnahme benutzt, um den Benutzern eine Vielzahl von Suchmethoden anzubieten, die das Retrieval und ihre Produkte verbesserten. Da diese Systeme verschieden aufgebaut waren, entwickelten Bibliotheken in den späten Achtziger Jahren ein Standard-Suchprotokoll, das eine übergreifende Suche über alle erlaubte, und in den Neunziger Jahren eine weit verbreitete Anwendung fand. Das war möglich aufgrund des hohen Standardisierungsgrades der Metadaten innerhalb des Bibliothekswesens, so dass das Protokoll sich auf andere Unterschiede konzentrieren konnte, die bei diesen historischen, pre-Web-Bemühungen Realität waren – unterschiedliche Basis-Architektur und unterschiedliche Ansätze für die Gestaltung. Heute ist dieses Protokoll Z39.50² eine wesentliche Komponente von Bibliothekskatalogen. Das Web-Interface der Library of Congress beispielsweise erhält zurzeit im Durchschnitt ungefähr 150.000 Suchanfragen an jedem Arbeitstag und mehr als die Hälfte davon kommen über das Z39.50-Protokoll.

Die Entwicklung und extensive Nutzung von Z39.50 hat die Tatsache unterstrichen, dass die Indexierung durch verschiedene Systeme ein wesentlicher Bestandteil der Suche in Online-Katalogen ist. Die falsche Zusammenführung von Indexierungen in verschiedenen Zielen kann eine große Verwirrung für den Endnutzer darstellen. Zu unterschiedlichen Zeiten wurden Anstrengungen unternommen, „Indices zu standardisieren“, allerdings hielt sich die Begeisterung der Systemverkäufer in Grenzen, manchmal lehnten sie es sogar ab. Während die Standardisierung der Metadaten selbst den Austausch von Metadaten ermöglicht, kann man argumentieren, dass die Freiheit, die reichhaltigen MARC-Metadaten zu indexieren, den Bibliotheken genutzt hat, weil sie die Systementwickler ermuntert, mit verschiedenen Ansätzen zur Suche zu experimentieren.

Wenn wir 20 Jahre weiter gehen, ins Jahr 2006, bietet die Technik viel mehr Möglichkeiten für ein plattform-übergreifendes Retrieval. In den Neunziger Jahren des letzten Jahrhunderts wurde das Web erfunden, und die Entwicklung von XML bot die Gelegenheit, das Verständnis für die Suche, das durch die Entwicklung und die Nutzung von Z39.50 aufgebaut wurde, auf diese neue Plattform zu übertragen. Seit 2000 sind eine Reihe von Gelegenheiten und Ansätzen aufgetaucht, die von den verfügbaren neuen Tools, Standards und der Unmenge an digitalem Volltextmaterial profitieren. Auf diese Weise werden die Komponenten des Retrieval – Metadaten, kontrolliertes Vokabular, Zusammenführung, Freitext-Stichwort, Indexierung, Wiederauffinden und Relevanz – in neuem Licht diskutiert.

Suche ist ein breites Thema, und dieser Vortrag legt den Schwerpunkt auf einige aktuelle Modelle und Protokolle, die zur systemübergreifenden Recherche propagiert werden. Insbesondere werden zwei Suchprotokolle untersucht, Search/Retrieve via URL (SRU) und OpenSearch/A9, sowie zwei Retrievalsprachen, Contextual Query Language (CQL) und XQuery. Die Metasuche bzw. Suche über verschiedene Sites selbst wird nicht behandelt, weil die meisten Client/Server-Protokolle so konfiguriert werden können, dass sie dem Endnutzer Metasuch-Ergebnisse liefern, wenn auch mit mehr oder weniger Anspruch und Transparenz.

Verwendete Terminologie. Der Vortrag unterscheidet eine Retrievalsprache von einem Suchprotokoll folgendermaßen: Ein Suchprotokoll ist eine Reihe von Nachrichten zwischen einem Client (der in diesem Vortrag *Benutzer* genannt wird) und einem Server (im Vortrag *Ziel* genannt). Der Benutzer ist in diesem Zusammenhang nicht der Endnutzer selbst, sondern sein System. Ein Bestandteil dieser Nachrichten ist die Abfrage selbst, aber sie enthält auch den Kontext der Abfrage, Informationen über den Benutzer und Einstellungen des Zieles, Informationen über das Ergebnis und das Handling des Ergebnis-Sets, welches das Ergebnis der Abfrage darstellt, und über das Vehikel zum Transport der gefundenen Datensätze (die Retrieval-Komponente). Der Begriff *Protokoll* wird im Allgemeinen für die Verwaltung des Datentransfers und des Retrievals benutzt und der Begriff *Abfrage* für die Argumentkomponente, welche die Suchkriterien enthält.

Eine *Retrievalsprache* ist die Syntax, die in einer strukturierten Form die Parameter für eine Abfrage festlegt, wie „suche nach dem Begriff ‚fish‘, der im Titel oder im Schlagwort vorkommt“. Eine wichtige Unterscheidung wird zwischen einer Abfrage in einer lokalen und/oder Benutzer-Syntax gemacht und einer, die eine abstrakte bzw. Standard-Syntax verwendet.

SRU – Search/Retrieve via URL

SRU ist ein Search- und Retrievalprotokoll, welches das Internet und die Möglichkeiten des Webs benutzt, um die Nachrichten zwischen dem Benutzer und dem Ziel zu transportieren. SRU hat einen bedeutenden Vorgänger, Z39.50, der weltweit sehr häufig angewendet wird. Vieles von der Funktionalität von SRU ist von dem älteren Protokoll abgeleitet, jedoch wurde nur das nützlichste übernommen und dazu in einer vereinfachten Form. Zu Beginn der Entwicklung von SRU wurde eine vergleichbare Suchanwendung der URL in verschiedenen Institutionen untersucht, insbesondere von Mitarbeitern in der Königlichen Bibliothek der Niederlande. Eine internationale Expertengruppe aus der Z39.50-Community arbeitete gemeinsam an dem Entwurf zu diesem neuen Protokoll für die Internet/Web/XML-Umgebung. Die SRU-Spezifikationen wurden 2002 erstmals veröffentlicht und werden gern bei neuen Anwendungen benutzt, weil sie einfach zu implementieren sind.³

SRU ist sehr flexibel. Es ist XML-basiert, und die bekannteste Implementierung ist SRU via URL, welche das HTTP GET zum Transfer der Nachrichten verwendet. Andere Versionen können allerdings über das Web-SOAP-Protokoll (SRU via SOAP) laufen, das mehr Web-Service-Funktionalitäten unterstützt, und über HTTP POST (SRU via POST), das kürzer ist und weniger Zeichensatzbeschränkungen hat als HTTP GET zurzeit. Die Datensätze, die als Antwort auf eine Suchanfrage ausgegeben werden, können in irgendeinem definierten XML-Format sein.

Das Z39.50- und das SRU-Modell

Das Funktionsmodell zur Suche in unterschiedlichen Seiten mit dem SRU-Protokoll entspricht dem für das Z39.50-Protokoll benutzten Modell. Der Endnutzer stellt eine Suchanfrage auf seinem (häuslichen) System, das eine spezifische lokale Frage-Syntax verwendet. Zur Suche in einem Zielsystem mit unterschiedlicher spezifischer Frage-Syntax, Datenbank-Design und Indexierungsregeln wird die lokale Suchanfrage in ein Standard-Format umgewandelt. Die Ziel-Site erhält diese Standard-Such-Nachricht, die aus Protokoll und Anfrage besteht, und übersetzt sie in die Syntax, die ihre Datenbank versteht. Wenn die

Anfrage sehr detailliert ist, oder die Suchparameter vom Zielsystem nicht unterstützt werden, kann das Zielsystem die Anfrage entweder als nicht möglich zurückweisen oder manchmal auch Merkmale, die es nicht bedienen kann, ignorieren bzw. verändern und dann mit der veränderten Suche fortfahren.

Der Protokollteil der Suche, der in der Abfrage enthalten ist, spezifiziert für das Zielsystem verschiedene Einstellungen: Format der gefundenen Datensätze, Anzahl der Datensätze, die angezeigt werden sollen, sowie andere Informationen für das Ergebnis etc. Das Protokoll unterstützt dann Antworten auf die Abfrage, wozu gefundene Datensätze, Fehleranzeige, Spezifikation des tatsächlichen Formats der angezeigten Datensätze etc. gehören können.

Ein Hauptproblem bei der Suche über unterschiedliche Systeme ist der Unterschied bei der Indexierung auf den verschiedenen Sites. Deshalb haben die Entwickler von Z39.50 und SRU zur Verbesserung der Qualität von Suchen über verschiedene Systeme eine Möglichkeit geschaffen, den Nutzer in die Lage zu versetzen, das Zielsystem zunächst zu fragen, was es unterstützt. Dabei wird vom Zielsystem erwartet, dass es die Antwort in einer standardisierten, im Protokoll spezifizierten Form ausdrücken kann. Man ist dabei davon ausgegangen, dass durch die Anwendung der erhaltenen Information über die Indices des Zielsystems eine erfolgreichere Suchanfrage vom System des Benutzers formuliert werden könnte.

SRU-Funktionen

SRU hat drei Basisfunktionen, die das oben genannte Modell ermöglichen: Explain (erläutern), Search/Retrieve (suchen/finden) und Scan (scannen).

SRU Explain. Aufgrund umfangreicher Erfahrungen mit Z39.50 behandelten die SRU-Entwickler die Explain-Funktion als ihre Priorität. Explain ist der Name der oben beschriebenen Möglichkeit, die dem Zielsystem gestattet, dem Benutzer zu sagen, was es machen kann. Das Ziel verfügt über eine Beschreibung seiner selbst in einem XML-Standardformat, die vom System des Benutzers leicht mittels einer Explain-Abfrage gefunden werden kann, und die dann dem Benutzer bei der Formulierung der Suchfragen helfen kann. Der Explain-Datensatz enthält eine Reihe von anderen beschreibenden Feldern, wie beispielsweise die XML-Formate, die für gefundene Datensätze geliefert werden können, oder die voreingestellte Zahl von Datensätzen, die wiedergegeben werden können. SRU-Ziele müssen nicht unbedingt, sollten aber möglichst, einen Explain-Datensatz unterhalten.

SRU Search/Retrieve. Die Search/Retrieve-Funktion ist die Hauptfunktionalität von SRU. Sie führt das Senden einer Suchanfrage zusammen mit der Protokollinformation über die Einstellungen an das Zielsystem durch und sorgt für die Antwort auf diese Anfrage, wie z.B. die Wiedergabe von Datensätzen, die zur Anfrage passen und dazu gehörige Informationen. Die verwendete Fragesyntax ist die unten beschriebene Contextual Query Language.

SRU Scan. SRU Scan versetzt den Anwender in die Lage, eine Ansicht der Suchbegriffe im Umfeld eines in einem Parameter spezifizierten Begriffs sowie die Anzahl der Treffer, die diesen Begriff enthalten abzufragen - natürlich nur, wenn das Zielsystem diese Browsingfähigkeit besitzt. Während die Search/Retrieve-Funktion es ermöglicht, Indices nach Begriffen für passende Datensätze zu durchsuchen, erlaubt die Scan-Funktion es dem Anwender, einen ganzen Bereich vorhandener Begriffe an einer bestimmten Stelle innerhalb des Indexes zu durchsuchen. Das ermöglicht es dem Anwender, eine geordnete Begriffsliste anzusehen und - sofern das unterstützt wird - zu sehen, wie viele Treffer es für eine Suche nach jedem einzelnen Begriff gäbe. Scan wird oft benutzt, um Begriffe für eine spätere Suche auszuwählen oder um ein negatives Suchergebnis visuell zu verifizieren.

Anwendungsbeispiel. SRU wird von Z39.50-Sites als zusätzlicher Zugang zu Katalogen verwendet und von Sites ohne Z39.50 als ein vereinfachter Zugang zu einer externen Suche. Die Library of Congress hat SRU im Jahr 2004 als eine Ziel-Site implementiert, indem sie ein Gateway zu Z39.50 verwendet, das Teil ihrer Katalog-Liefersoftware ist, anstatt direkt an die Datenbanken zu gehen. Diese Software, die SRU einschließt, erlaubte es der LC auch, ihre direkten Z39.50-Antworten zu verbessern und anzubieten, Datensätze in MARC 21,

MARCXML, MODS und sogar DC auszugeben. Die Bibliothek ist dabei, SRU in einer Reihe anderer Datenbanken zu implementieren, die nicht von Haus aus MARC-orientiert sind, allerdings geht man davon aus, dass MARCXML und MODS bei den Antworten angeboten werden.

CQL – Contextual Query Language

Eine Schlüsselkomponente der Funktion Search/Retrieve ist die Abfrage. Die Schöpfer von SRU haben eine Abfrage-Syntax entwickelt, die sowohl reichhaltig als auch einfach ist - und gut geeignet -, das Optimum aus Bibliotheks-Metadaten herauszuholen. Diese Retrievalsprache ist die Contextual Query Language bzw. CQL, wie sie normalerweise genannt wird.⁴

CQL ist eine formale Sprache zur Wiedergabe von Anfragen. Sie wurde entworfen, um Systeme zum Information Retrieval, wie Web-Indices, bibliografische Kataloge und Museumsbestandsinformationen einander anzupassen. Im Gegensatz zu der Suchsyntax, die normalerweise bei Z39.50 verwendet wird, soll CQL von Menschen gelesen und geschrieben werden können und angemessen intuitiv sein, obgleich bedeutende Bestandteile der hoch expressiven (und komplexen) Z39.50-Retrievalsprache beibehalten werden. Dadurch ist sie mächtiger als eine einfache Google-artige Sprache. Wie auf der Website für SRU angemerkt wird, „versucht CQL die Schlichtheit und Intuitivität des Ausdrucks für einfache Alltagsanfragen mit dem Reichtum von expressiveren Sprachen zu kombinieren, um komplexe Begriffe einander anzugleichen, wenn dies notwendig ist.“⁵ Obwohl CQL anstrebt, für Menschen lesbar zu sein, geht man dennoch davon aus, dass die Endnutzer eine Oberfläche haben, die ganz einfach ihre lokale Katalogoberfläche und Syntax sein kann.

CQL basiert auf der Definition eines Sets von abstrakten Sucheinstiegen, wie z.B. Titel, Verfasser, Thema, und deren Verfeinerungen, wie persönlicher Autor, Einheitstitel, geografisches Thema. Während große Datenbanken in der Regel eine Art Indexstruktur aufweisen und die abstrakten Sucheinstiege bei der CQL oft abstrakt als „Indices“ bezeichnet werden, erfordert CQL nicht tatsächlich die Existenz von „physischen“ Indices beim Ziel, sondern die Fähigkeit zum Retrieval, als ob es welche gäbe. CQL erfordert kein bestimmtes Datenbankdesign – egal ob die Datenbank relational, objektorientiert, hierarchisch, vernetzt etc. ist –, sondern ist darauf ausgerichtet, identifizierte Metadaten zu suchen (d.h. Datensätze als Daten und nicht als Dokumente), um „intelligentes“ Suchen zu ermöglichen. Auf diese Art können zwar einfache Suchanfragen unterstützt werden, aber die Stärken werden nicht genutzt.

Ein anderes Modell – OpenSearch

Im März 2005 hat die Firma Amazon einen neuen Dienst vorgestellt, den sie OpenSearch genannt hat. Es handelt sich dabei um eine Implementierung eines Suchprotokolls, das von der Software-Firma A9⁶ entwickelt wurde. Während OpenSearch/A9 mit einer Reihe von unterschiedlichen Diensten, wie z.B. RSS Feeds, zusammenhängt, liegt das Augenmerk hier auf der Komponente Suchprotokoll – auf dessen funktionellem Modell, darauf wie es sich von SRU unterscheidet, auf den komparativen Vorteilen und der möglichen Kombination beider Ansätze.

Ähnlich wie die Modelle im Bibliotheksbereich hat Amazon erkannt, dass „traditionelle Suchmaschinen oftmals nicht so gut den Inhalt von fachlichen Websites indexieren und dass lokale Suchmaschinen lokale Inhalte weitaus besser verstehen können: ‚Unterschiedliche Arten von Content erfordern unterschiedliche Arten von Suchmaschinen. Und meist sind die besten Suchmaschinen für eine Site diejenigen, die von denjenigen entworfen werden, die den Content am besten kennen.“⁷ Dies ist ein Anerkennen, dass z.B. ein Katalog mit medizinischer Information wahrscheinlich ein hoch spezialisiertes und präzises Search und Retrieval bietet, und dass sogar in Datenbanken mit allgemeinem Material die Suche an einer bestimmten Klientel orientiert sein kann, wie z.B. Schulkinder, Forscher oder

Kunststudenten. Folglich kann es funktionelle Unterschiede bei der Implementierung von Suchen in Dateien geben. Es ist im Wesentlichen das Problem der unterschiedlichen Indexierung, das von den anderen externen Suchprotokollen wie Z39.50 und SRU erkannt wurde, und wofür diese versuchen, Lösungen anzubieten.

In ihrem Bemühen, fachliche Suchimplementierungen anzuzapfen, entwickelten Amazon und A9 das folgende Modell. Das System des Benutzers schickt eine Abfrage nach Informationen über dessen Lokalsystem an das Ziel. Das Ziel schickt die Namen der Parameter zurück, die lokal für Suchaktivitäten verwendet werden. Der Benutzer schickt dann eine Abfrage an das Ziel, wobei er die „Sprache“ des Ziels benutzt, und erhält die gefundenen Datensätze von dem Ziel im RSS-Format. Während das zugrunde liegende Konzept interessant ist und die Interaktionen überwiegend mit SRU kompatibel sind, macht OpenSearch es zwar sehr leicht, eine einfache Stichwort-Suche durchzuführen, aber bedeutend schwieriger, etwas mit größerer Präzision zu machen.

Der OpenSearch-Client fragt zunächst minimale Informationen über das Ziel ab – besonders die Namen der Parameter, die für das lokale Suchformular des Ziels benutzt werden müssen. Die Besonderheiten der Sucheinstiege, die durch das Ziel unterstützt werden, werden nicht angesprochen. Man braucht keine Standard-Suchsyntax wie CQL, um verfeinerte Abfragen zu übermitteln, weil die Abfragen nur Stichworte sein können. Es gibt auch keine Auswahl der Syntax der wiedergegebenen Datensätze.

OpenSearch ist nützlich, wenn es um ein Suchprotokoll mit einer sehr niedrigen Schwelle geht, das in erster Linie die Stichwortsuche unterstützt. Es akzeptiert die Tatsache, dass in unterschiedlichen Datenbanken verschiedene Suchen definiert sein können und benutzt einfach eine Stichwortsuche, die nach den Vorgaben des Ziels bearbeitet wird. Man nimmt dabei an, dass der Endnutzer nicht wissen muss, wie der Begriff vom Ziel verarbeitet wird.

Der Unterschied zwischen OpenSearch und SRU mag darin liegen, dass OpenSearch nicht versucht, das Ziel zu verstehen, sondern einfach nur versucht, das Ziel in die Lage zu versetzen, den Suchbegriff als Stichwort zu behandeln. SRU kann solche einfachen Suchen unterstützen, aber versucht auch mit der Funktion Explain und der Entwicklung von abstrakten Sucheinstiegen, das Ziel zu verstehen und die Besonderheiten der Abfrage seines Benutzers zu interpretieren, um die präziseren Ergebnisse zu erhalten, die dieser möchte. SRU unterstützt auch den Wunsch nach einer spezifischen XML-Syntax für die wiedergegebenen Datensätze, während OpenSearch sein Protokoll durch die Beschränkung der gefundenen Information auf die RSS-Syntax vereinfacht hat.

Der Ansatz von OpenSearch ist besonders wertvoll zur Suche über viele unstrukturierte Server und Dateien. Obwohl die Ergebnisse heterogen sein können, sind sie für einige Zwecke geeignet. Auch als Testrecherche in Sites, die dann im Anschluss direkt durchsucht werden können. SRU benutzt man am besten zur Suche in hinreichend strukturierten Katalogen, wenn ein Benutzer eine besser strukturierte Suchmöglichkeit und mehr Kontrolle über die Ergebnisse wünscht. Suchende, die an einen lokalen Bibliothekskatalog gewöhnt sind, werden SRU aufgrund seiner Ähnlichkeit mit der lokalen Suche nützlicher finden. Suchende, die von einer Google-Suche kommen, finden die OpenSearch-Ergebnisse möglicherweise zufrieden stellend.

Die zwei Protokolle sind kompatibel und zwischen A9- und SRU-Entwicklern haben erste Gespräche stattgefunden⁸. Ziel soll sein, OpenSearch in die Lage zu versetzen, umfassender zu suchen sowie die Lücke zwischen Umgebungen zu überbrücken, die zwar nicht das etwas genauere SRU, aber dennoch mehr Funktionalitäten benötigen als OpenSearch gegenwärtig bietet.

XQuery

XQuery ist eine Abfrage-Syntax für XML, die vom World Wide Web Consortium (W3C) entwickelt wird⁹. Die Arbeit ist in den vergangenen 7 Jahren nur langsam fortgeschritten durch eine grundlegende Uneinigkeit zwischen denjenigen, die XML in erster Linie als Dokumenten-Markup-Sprache (zur Identifizierung von Absätzen, Kapitelanfängen etc.)

ansehen, und denen, die XML als Daten-Markup-Sprache (zur Identifizierung von Namen, Schlagworten und Daten) bevorzugen. Sie ist abhängig von der Kenntnis der XML-Tags in den Dokumenten, die sie durchsucht, und ist darauf ausgerichtet, mit den Basisdaten und nicht mit den Indices zu arbeiten. Aber XQuery ist eine hoch komplexe (und umfassende – wenn in diesem Bereich überhaupt etwas umfassend sein kann) Abfragesprache und dürfte ihren Hauptnutzen eher im Bereich der Steuerung von Abfragen innerhalb eines Systems haben als in Client/Server-Applikationen. SRU und OpenSearch, die beide Client/Server-Protokolle sind, könnten mit XQuery-Applikationen kommunizieren, wenn sie den Server, so wie sie es heute tun, mit SQL (Structured Query Language) oder einer anderen Datenbank-Suchsoftware erreichen.

Schlussfolgerung

Dieses Papier untersucht einige der führenden „Akronyme“, die zurzeit in der Client/Server-Informationssuchumgebung populär sind – Suchprotokolle wie SRU und OpenSearch/A9 und Abfragesprachen wie CQL und XQuery. Bibliotheken brauchen die unterschiedlichen Ansätze – „intelligentes“ Suchen von strukturierten Daten in Katalogen und unstrukturiertes Suchen von digitalisierten Dokumenten und Katalogdateien. SRU und OpenSearch sind innovative und überdachte Suchlösungen für die Vielfalt der Suchsysteme, die von Serverzielen angeboten werden. Die Zusammenarbeit ihrer Entwickler ist eine willkommene Nachricht und wird hoffentlich zu einer Verbesserung beider Ansätze führen. In der Zwischenzeit bietet CQL – heute – eine überprüfte und angemessen strukturierte Suchsyntax für Client/Server-Applikationen, die auf eine Vielzahl von Server-Eigenheiten anwendbar ist, während das umfassende und hoch komplexe XQuery eine Grundlage für zukünftige Entwicklungen der Suche liefern kann.

Übersetzt von Susanne Oehlschläger, Deutsche Nationalbibliothek

Anmerkungen

¹ Die Verfasserin dankt Ray Denenberg, Library of Congress, und Robert Sanderson, University of Liverpool für Untersuchungen und Informationen zu diesem Papier. Fehler gehen auf mein Konto.

² Die Web-Site des Z39.50-Protokolls ist unter <http://www.loc.gov/z3950/agency/> zu finden. Auf der Seite gibt es Links zur ANSI/NISO-Version des Standards, und eine gedruckte Ausgabe des Standards (ISO 23950) ist bei ISO erhältlich.

³ Unter <http://www.loc.gov/standards/sru> ist die SRU-Website zu finden. Die SRU-Spezifikation, Beschreibungen der Entwicklung und Hilfsdokumente sind auf dieser Site erhältlich.

⁴ Die CQL-Website ist unter <http://www.loc.gov/standards/sru/cql> zu finden.

⁵ <http://www.loc.gov/standards/sru/cql> (Zugriff am 30. Mai 2006)

⁶ Wiggins, Richard W., „Amazon’s New OpenSearch Enables Search Syndication“, March 28, 2005 <<http://www.infotoday.com/>> (Zugriff am 22. April 2006)

⁷ From the OpenSearch FAQ located at <<http://opensearch.a9.com/docs/faq.jsp>> (Zugriff am 22. April 2006)

⁸ „Rob Sanderson visits A9.com“

<<http://blog.a9.com/blog/2006/04/14/rob-sanderson-visits-a9com/>> (Zugriff am 29. Juni 2006)

⁹ „XML Query (XQuery) Requirements“, W3C Working Draft 3 June 2005

<<http://www.w3.org/TR/xquery-requirements>> (Zugriff am 30. Mai 2006)