

S protokolom Z39.50 je to jinak – není to žádný jazyk, je to soustava protokolů sloužících ke komunikaci mezi heterogenními knihovnickými systémy. Umožňuje (s určitými výhradami) vyhledávat, editovat (updatovat), up- a downloadovat záznamy. Ke své činnosti používá další knihovnické standardy, např. standardy typu MARC a ISO 2709 a řadu jiných.

Komunikace prostřednictvím Z39.50 mezi dvěma knihovnickými systémy probíhá (např. v případě akce update záznamu v knihovnickém systému B z prostředí knihovnického systému A) takto: Z39.50 updatovací klient systému A (představujme si např., že B je souborný katalog a systém A je systém spolupracující lokální knihovny) naváže spojení se Z39.50 serverem systému B a ten bude nadále zprostředkovávat předávání všech editačních informací systému B. Systém B tak zpřístupní editovaný záznam klientu systému A a bude provádět jeho editační příkazy. Katalogizátor bude reálně pracovat v systému B, ale bude používat prostředí svého lokálního systému A. Při komunikaci musí A klient Z39.50 a B server Z39.50 používat stejné rozhraní – to, co jeden vyšle, musí druhý přijmout (proto použijí např. standard ISO 2709) a navíc oba musí předávané informace stejně interpretovat (proto použijí např. UNIMARC). Ve skutečnosti je k úspěšné komunikaci potřeba víc: pro zjednodušení jsme vynechali např. problematiku profilů a verzí protokolu Z39.50, problematiku katalogizačních pravidel...

V tomto okamžiku už by mělo být vzájemné postavení protokolu Z39.50 a jazyka XML jasné. Mohou a asi i budou existovat vedle sebe a spolu – např. v kombinaci XML/UNIMARC mohou nahradit dvojici ISO 2709/UNIMARC, a to nejen při komunikaci prostřednictvím Z39.50.

Poslední verzi tohoto pojednání četlo několik mých spolupracovníků a spolupracovníků, děkuji jim za připomínky. Na několik nepřesností mě upozornil kolega Aleš Horák, tomu patří mé poděkování nejvíce.

Použitá literatura:

BRADLEY, Neil. *XML : kompletní průvodce*. Přeložil Jiří Brázda. 1. vyd. Praha : Grada, 2000. 537 s.

KOSEK, J. *XML pro každého : podrobný průvodce*. 1. vyd. Praha : Grada 2000. 163 s.



Z39.50 A XML

Nadežda Andrejčíková

Cosmotron Bohemia s.r.o., Hodonín

V súvislosti s kooperáciou sa s týmito dvomi označeniami stretávame stále častejšie. Pripomína mi to tak trochu situáciu spred pár rokov, keď bol u nás prijatý medzinárodný štandard Unimarc, s ním súvisiace Unimarc Authority a pravidlá AACR2. V rozhovoroch knihovníkov sa neustále skloňovali tieto pojmy, i keď zo začiatku nebolo hneď všetkým jasné, čo sa za nimi skrýva. Vývojová špirála ide dopredu, mnohé knižnice úspešne prešli automatizáciou, elektronizáciou a niektoré sú už dokonca v určitom štádiu digitalizácie. A práve tu sa dostávajú na rad nové pojmy Z39.50 a XML. Čo sa za nimi skrýva? K čomu slúžia? Ako nám môžu pomôcť? Môžeme ich navzájom zamieňať? Tieto a mnohé ďalšie otázky si položili mnohí z nás a názory na ne nie sú vôbec rovnaké. Skôr, ako sa pokúsím vyjadriť svoj názor na tieto otázky, je potrebné, aby sme si povedali, čo vlastne Z39.50 a XML je.

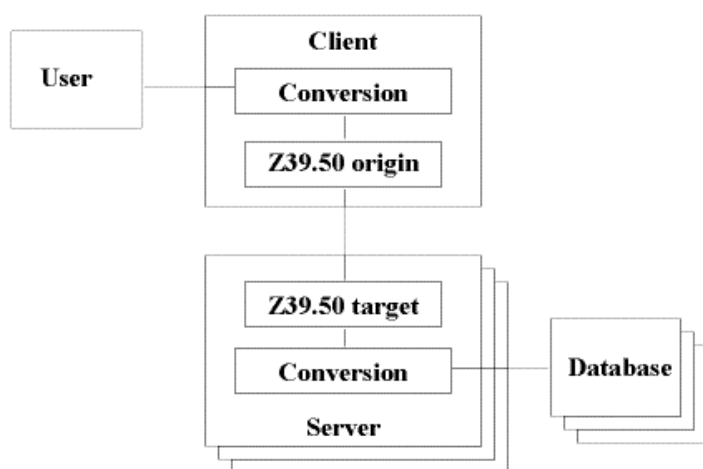
Z39.50

Mohla by som začať citáciou definície tohto protokolu priamo z normy, ale myslím, že to vôbec nie je potrebné. Definíciu i komplexný popis tohto protokolu si môže každý nájsť na stránkach LOC (Library of Congress – www.locweb.loc.gov/Z39.50/agency). To, že Z39.50 je komunikačný protokol určený pre dorozumievanie sa informačných systémov (IS) s cieľom vyhľadávania a preberania informácií nie je potrebné zdôrazňovať. Treba však zdôrazniť, že bol pôvodne prijatý ako americký národný štandard, ktorý bol potom prijatý i medzinárodnou organizáciou pre štandardy ako ISO norma. Súčasná verzia označovaná ako verzia č. 3 bola prijatá ako norma ISO 23950. Takže ide o medzinárodný štandard, ktorý je určený pre IS ako také, v ktorých bibliografické informačné systémy tvoria len jednu časť. Tento protokol môžeme charakterizovať tromi základnými prvkami:

1. abstraktný model pre vyhľadávanie a prenos informácií
2. jazyk, v ktorom je presne definovaná syntax a sémantika pre vyhľadávanie a prenos informácií tak, aby umožnil komunikáciu IS
3. predpisuje spôsob kódovania požiadaviek i odpovedí, ktoré sú potom posielané v rámci sieťovej infraštruktúry.

Znamená to teda, že komunikovať tak môžu medzi sebou IS, ktoré majú nad sebou implementovaný tento protokol. Samozrejme nič nie je také jednoduché, ako na začiatku vyzerá. Norma je veľmi všeobecná. Jej najväčšou nevýhodou, ale na druhej strane aj výhodou je to, že ide o abstraktný model, teda model, ktorý nie je pevne viazaný na žiadny systém a ani na žiadnu databázu. K tomu, aby si IS navzájom porozumeli, je potrebné implemento-

vať tento protokol na strane servera, v norme je tento označovaný ako *target*, i klienta, označovaného ako *origin*. Klient i server medzi sebou komunikujú v tzv. Z-jazyku s pevne definovanou syntax a sémantikou. V prípade, že koncový používateľ nekomunikuje priamo cez Z-klienta, tak sa klient stará o preklad jeho požiadaviek do tohto jazyka. Takto definované PDU (Protocol Data Unit), potom kóduje na základe pravidiel označovaných ako BER a to potom posieľa cez transportnú vrstvu, ktorou môže byť napríklad TCP, na vzdialený Z-server. Tento danú požiadavku dekóduje a preloží do jazyka reálnej databázy, ktorá ju vykoná a výsledok pošle späť serveru. Ten opäť tento výsledok preloží do svojho jazyka, zakóduje a pošle klientovi, kde sa deje opačný proces, tak, aby odpovedi porozumel používateľ. Graficky môžeme architektúru klient server Z39.50 vyjadriť nasledovne:



Avšak samotná implementácia tohto protokolu nezaručuje 100%-nú kompatibilitu IS. Server i klient musia mať implementované rovnaké funkcie s danými parametrami. To, ktoré funkcie sú implementované na strane klienta i servera je uvedené v tzv. profile. Najjednoduchším profilom je tzv. ATS-1 profil umožňujúci prehľadávať jednotlivé databázy podľa autora, titulu a subjectu, z čoho je odvodený aj jeho názov. Pre bibliografické IS je definovaných viacero profilov. V rámci medzinárodnej kooperácie sa doporučuje tzv. BATH profile.

XML

S jazykom XML (eXtensible Markup Language) sa v počítačovom svete stretávame stále častejšie. Môžeme povedať, že oveľa viac počítačových odborníkov vie niečo o XML, ako o Z39.50. Mnohí si od tohto jazyka veľa sľubujú, i keď na druhej strane tých, ktorí skutočne vedia podstatu XML je pomerne málo. O XML sa väčšinou hovorí v súvislosti s tvorbou webovských stránok a teda aj v súvislosti s jazykom HTML. Práve s jazykom HTML býva často porovnávaný a skutočne s ním má aj veľa spoločného. Avšak na druhej strane XML na rozdiel od HTML si môže definovať vlastné značky (tagy). Na základe toho

je možné označiť význam jednotlivých častí textu. Pričom HTML umožňuje len definovať vzhľad textu pri jeho zobrazení. Prvým takýmto podobným značkovacím jazykom bol GML (Generalized Markup Language), ktorý sa natoľko osvedčil, že na jeho základe vznikol SGML (Standard Generalized Markup Language), ktorý je tiež definovaný v ISO norme ISO 8879. Tento je tak obecný, že pomocou DTD (Document Type Definition) umožňuje definovať vlastné značkovacie jazyky. Preto i HTML a XML vznikli na jeho základe. XML môžeme teda definovať ako podmnožinu SGML s možnosťou definovať si vlastné DTD. Výhodou XML je to, že má oveľa presnejšie definovanú syntax, čo ho zvyhodňuje v rýchlosti a cene v ňom vyvíjaných aplikáciách. XML bol vyvinutý za účelom uchovávanía a spracovávanía dokumentov a na to sa aj hlavne zameriava. Pomocou XML značiek môžeme v dokumente označiť, že toto je názov, toto autor atď. Teda

XML dokumenty obsahujú oveľa viac informácií, ako keby sa používalo len prezentačné označovanie. Informačná bohatosť XML dokumentov sa dá využiť vo viacerých oblastiach. Snáď najväčším prínosom bude vyhľadávanie, pretože dnešné internetové prehliadače, tak ako je napr. AltaVista, podporujú len fultextové vyhľadávanie. Pričom, ak by sme použili XML a zadali slovo, ktoré chceme hľadať, a definovali pritom, že toto slovo sa má nachádzať v názve dokumentu, bude prehľadávanie oveľa presnejšie a výsledok relevantnejší našej požiadavke. Ďalšou výhodou použitia XML je jeho pomerne ľahká konverzia do iných formátov. I keď XML v sebe neobsahuje žiadne značky pre to, ako budú jednotlivé časti dokumentu zobrazené napr. na obrazovke, je to možné vďaka hneď niekoľkým tzv. štýlovým jazykom (XSL – eXtensible Stylesheet Language). Jeden XML dokument môže mať definovaných hneď niekoľko takýchto formátov. V čom je teda sila XML a kde všade ho môžeme použiť?

Sila XML je najmä v jeho hierarchickej štruktúre a pomerne jednoduchom spôsobe kódovania. Umožňuje popisovať – označovať ľubovoľné dáta a prenášať ich medzi rôznymi aplikáciami a platformami. Hlavnou ideou XML je oddelenie obsahu a dizajnu dát. Spojovacím mostom je XSL, ktorý umožňuje vytvárať množstvo výstupných formátov.

Otázkou je, či teda môže XML nahradiť protokol Z39.50, keď aj XML umožňuje prenos dát medzi IS po sieti. Odpoveď na túto otázku nie je tak jednoduchá. Môj názor je ten, že XML môže byť úspešne využité v rámci protokolu Z39.50, ale nemyslím si, že by ho mohol v krátkej budúcnosti úplne nahradiť. V stručnosti sa pokúsím načrtnúť, ako by mohol byť XML využitý v Z39.50. Tak, ako som charakterizovala Z39.50 v úvode tohto článku v troch úrovniach, tak si myslím, že o XML môžeme hovoriť v súvislosti s druhým a tretím bodom. Teda v rámci syntaxe a kódovania prenášaných údajov po sieti. Sémantika je daná samotným Marc formátom, v ktorom protokol Z39.50 komunikuje. Je možné aj tento nahradiť XML, ale

nemalo by to asi podstatný význam, pretože by musela byť spätná konverzia na strane aplikácií. Nemyslím si však, že by XML mohol nahradiť protokol Z39.50 v oblasti PDU, ktoré sú v tomto protokole veľmi dobre definované. To by museli byť definované všetky funkcie a parametre i pre XML v rámci určitého štandardu tak, aby boli zrozumiteľné všetkým systémom, čo môže trvať niekoľko rokov, a otázkou je, v akom rozsahu by sa to uplatnilo. Otázka syntaxe a hlavne otázka kódovania údajov je však oveľa ľahšie aplikovateľná a môže znamenať určitý prínos. Tak ako som uviedla vyššie, požiadavky klienta (PDU definované v ASN – abstrakt syntax notation) sú pred transportom po sieti kódované na základe BER kódovania a tak sú pre používateľa veľmi ťažko čitateľné a zrozumiteľné. Pre ilustráciu si môžeme uviesť ako vyzerá PDU v ASN.1 pre update databáz, ktorý je definovaný v 3 verzii protokolu Z39.50.

ESFormat-Update

```
{Z39-50-extendedService Update (5) revisions (1)
  revision-1 (1)} DEFINITIONS ::=
  — oid is 1.2.840.10003.9.5.1.1
```

BEGIN

IMPORTS DiagRec, InternationalString

FROM Z39-50-APDU-1995;

Update ::= CHOICE {

```
  esRequest [1] IMPLICIT SEQUENCE {
    toKeep [1] OriginPartToKeep,
    notToKeep [2] OriginPartNotToKeep},
  taskPackage [2] IMPLICIT SEQUENCE {
    originPart [1]
      OriginPartToKeep,
    targetPart [2] TargetPart}}
```

OriginPartToKeep ::= SEQUENCE {

```
  action [1] IMPLICIT INTEGER {
    recordInsert (1),
    recordReplace (2),
    recordDelete (3),
    elementUpdate (4),
    specialUpdate (5)},
  databaseName [2] IMPLICIT InternationalString,
  schema [3] IMPLICIT OBJECT IDENTIFIER
    OPTIONAL,
  elementSetName [4] IMPLICIT InternationalString
    OPTIONAL,
  actionQualifier [5] IMPLICIT EXTERNAL
    OPTIONAL}
```

OriginPartNotToKeep ::= SuppliedRecords

TargetPart ::= SEQUENCE {

```
  updateStatus [1] IMPLICIT INTEGER {
    success (1),
    partial (2),
    failure (3)},
  globalDiagnostics [2] IMPLICIT SEQUENCE OF
    DiagRec OPTIONAL,
```

— These are non-surrogate
— diagnostics relating to the task,
— not to individual records.

taskPackageRecords [3] IMPLICIT SEQUENCE OF
TaskPackageRecordStructure

— There should be a
— TaskPackageRecordStructure
— for every record supplied.
— The target should create
— such a structure for every
— record immediately upon
— creating the task package
— to include correlation
— information and status.
— The record itself would not
— be included until processing
— for that record is complete.

}

— Auxiliary definitions for Update

SuppliedRecords ::= SEQUENCE OF SEQUENCE {
recordId [1] CHOICE {

```
  number [1] IMPLICIT INTEGER,
  string [2] IMPLICIT InternationalString,
  opaque [3] IMPLICIT OCTET STRING}
  OPTIONAL,

```

supplementalId [2] CHOICE {

```
  timeStamp [1] IMPLICIT GeneralizedTime,
  versionNumber [2] IMPLICIT InternationalString,
  previousVersion [3] IMPLICIT EXTERNAL}
  OPTIONAL,

```

correlationInfo [3] IMPLICIT CorrelationInfo OPTIONAL,
record [4] IMPLICIT EXTERNAL}

CorrelationInfo ::= SEQUENCE {

— origin may supply one or both for any record:

```
  note [1] IMPLICIT InternationalString OPTIONAL,
  id [2] IMPLICIT INTEGER OPTIONAL}
```

TaskPackageRecordStructure ::= SEQUENCE {

```
  recordOrSurDiag [1] CHOICE {
    record [1] IMPLICIT EXTERNAL,
    — Choose ,record' if
    — recordStatus is ,success', and
    — elementSetName was supplied.
```

```
  surrogateDiagnostics [2] IMPLICIT
    SEQUENCE OF DiagRec
```

```
  — Choose ,SurrogateDiagnostics', if
  — RecordStatus is failure.
  } OPTIONAL,
```

— The parameter recordOrSurDiag
— will thus be omitted only if
— ,elementSetName' was omitted and
— recordStatus is ,success'; or
— if record status is ,queued'
— or in ,process'.

```
  correlationInfo [2] IMPLICIT
    CorrelationInfo OPTIONAL,
```

— This should be included
 — if it was supplied by the origin.

```
recordStatus [3] IMPLICIT INTEGER {
    success (1),
    queued (2),
    inProcess (3),
    failure (4)},
supplementalDiagnostics [4] IMPLICIT
SEQUENCE OF DiagRec OPTIONAL}
```

END

Ak by sa toto kódovanie nahradilo kódovaním v XML na základe určitých pravidiel, boli by prenášané dáta oveľa čitateľnejšie. Tiež XSL by umožnilo rozšíriť definované formáty zobrazenia prenášaných dát, pričom by mohla byť zachovaná spätná konverzia do Marc formátu. O tom, že začlenenie XML do Z39.50 je reálne, svedčí aj to, že sa tejto problematike intenzívne venujú aj členovia ZIG-u (skupina implementátorov protokolu Z39.50). Na jednom

z ich stretnutí v roku 2000 Ray Denenberg diskutoval k hlavným cieľom XML protokolu. Jeho prednášku môžete nájsť na stránke: <http://lcweb.loc.gov/z3950/agency/zig/meetings/dc2000/presentations/xp.ppt>.

Ak by sme teda chceli znázorniť začlenenie XML do Z39.50, budeme hovoriť o ZX-klientovi a ZX-serveri. Aby bola zabezpečená komunikácia aj medzi klasickými Z-klientami a ZX-serverom, ako aj komunikácia medzi ZX-klientom a Z-serverom, je potrebné definovať bránu, ktorú môžeme označiť ako ZX-gateway, alebo XZ-gateway. Výhodou začlenenia XML do protokolu Z39.50 je možnosť využitia rôznych transportných vrstiev, ale tiež pre užívateľa oveľa jednoduchšie a čitateľnejšie XER kódovanie, ktoré by nahradilo BER kódovanie. Toto môžeme znázorniť nasledujúcou schémou.

Čo povedať na záver? XML je určite veľmi silný a dôležitý nástroj. Avšak ako všetko nové, tak aj XML potrebuje svoj čas na to, aby si našlo svoje miesto v IT/IS. Verím tomu, že v kombinácii so Z39.50 to bude silný nástroj, ktorý ešte viac zblíži jednotlivé IS a umožní im vzájomné zdieľanie ich dát.

