

УДК 025.3 MARC:004.43.057.2/3 XML

MARC ЗАПИСИ И XML

Гордана Будимир
Институт информацијских знаности
Марибор, Словенија

Сажетак

У овом чланку описан је развој формата MARC записа од настанка до данас - од ISO 2709 до XML формата. XML формат је општи формат којим се MARC записи могу представити на једноставан и разумљив начин. Алати за рад са XML форматом углавном су декларисани као отворени код (*open source*) и могу се искористити за обраду XML записа без додатних трошкова за развој сопствених решења. XML формат је главни формат за пренос података преко Интернета и Web сервиса. Међутим, док год буду постојали рачунарске системи са старим технологијама, постојаће и MARC записи у ISO 2709 формату.

Кључне речи: MARC, формати, ISO 2709, SGML, XML

1. Увод

Први библиографски формати за размену библиографских и њима сродних података, у машински читљивом облику, између различитих рачунарска система развијени су пре више од 30 година. У Конгресној библиотеци (Library of Congress) [1] тада су развили библиографски формат MARC (Machine Readable Cataloguing) [2], на основу којег су, за потребе различитих држава и библиотечких система, касније развијени други библиографски MARC формати, нпр.: USMARC у Америци, CAN/MARC у Канади, из којих је 1999. године настао формат MARC21 [3], UKMARC [4] у Великој Британији, UNIMARC (Universal Machine Readable Cataloguing) [5] за потребе европских држава, итд. На основу UNIMARC формата развијени су и неки национални библиографски формати [6].

MARC формат одређује садржај MARC записа чију структуру, са логичког аспекта, чине поља која су идентификована троцифреним таговима и груписана у блокове. Поља MARC записа садрже највише два индикатора и одређен скуп потпоља, дефинисан самим MARC форматом. MARC форматом одређује се и поновљивост поља у запису и поновљивост потпоља у пољима библиографског записа. Правила каталогизовања могу бити различита за различите MARC формате: ISBD (*International Standard Bibliographic Description*) [7], AACR (*Anglo-American Cataloguing Rules*) [8], LCSH (*Library of Congress Subject Headings*) [9] и слично.

UDC 025.3 MARC:004.43.057.2/3 XML

MARC RECORDS AND XML

Gordana Budimir
The Institute of Information Sciences,
Maribor, Slovenia

Abstract

This article describes how a MARC record has evolved in the decades since its introduction, i.e. from the ISO 2709 to XML format. The XML format is a general format for presenting MARC records in a simple and understandable way. Tools that work with the XML format are of the open source type and can be used to process XML records without incurring additional costs associated with one's own software development. XML is also the main format for data transfer on the Internet and for Web services. It is evident though that as long as computer systems using old technology exist, MARC records in the ISO 2709 standard format will also survive.

Key words: MARC, formats, ISO 2709, SGML, XML

1. Introduction

First bibliographic formats for the exchange of bibliographic and related information in the machine-readable form between different types of computer systems were created more than 30 years ago. Then the *Library of Congress* [1] created MARC (*Machine Readable Cataloguing*) [2] bibliographic format, on which other versions of MARC format were developed to meet the needs of a variety of countries and bibliographic systems, such as: the USMARC in the USA, the CAN/MARC in Canada, from which MARC21 [3] format was developed in 1999, the UKMARC [4] in the United Kingdom, the UNIMARC (*Universal Machine Readable Cataloguing*) [5], slightly adapted for the use in the European countries, etc. Based on the UNIMARC format, some national bibliographic formats have been created [6].

The MARC format specifies the content of MARC records, the structure of which, from logical point of view, consists of fields that are identified by three-character numeric tags and arranged in functional blocks. MARC record fields consist of two indicators at the most and of a set of subfields that are defined by the MARC format. Also, the repeatability of fields in a record and the repeatability of subfields in bibliographic record fields are defined by the MARC format. The cataloguing rules may differ for different MARC formats: ISBD (*International Standard Bibliographic Description*) [7], AACR (*Anglo-American Cataloguing Rules*) [8], LCSH (*Library of Congress Subject Headings*) [9], etc.

Са физичког аспекта структура MARC записа може бити различита. Тако су од појава MARC формата били дефинисани различити формати физичке структуре MARC записа, који су хронолошки представљени у овом чланку. Од појаве спецификације XML (*eXtensible Markup Language*) [10] дефинисање формата за библиографске записе добија нову димензију. XML није један нови формат библиографских записа већ ова одговарајућа спецификација XML формата библиографских записа омогућава лакшу обраду библиографских записа у библиотечким информационим системима. За имплементацију функција, као што су валидација записа, или трансформација записа у различите структуре, постоје већ готови алати, развијени у Java технологији, који су углавном декларисани као отворени код (*open source*).

2. ISO 2709 записи

У време настанка MARC формата развој рачунара није био на данашњем нивоу нити је њихова употреба била прилагођена обичним корисницима као што је данас. У то време није било развијених стандарда за размену података који би омогућавали приказ података у једноставним, човеку разумљивим, форматима. Једини физички носилац података који је омогућавао пренос података са једног рачунара на други била је магнетна трака. Подаци на магнетној траци били су уписани секвенцијално и тако су морали бити и читани. Зато су се подаци на траку паковали тако да су на почетку сваког записа груписане информације које су описивале његов садржај, нпр.: дужина записа, статус записа, имена поља у запису, итд. У складу са овим захтевима, стандард ISO 2709 (*Формат за размену информација*) [11] формиран је за размену библиографских MARC и њима сродних записа.

Стандард ISO 2709 описује начин преноса поља MARC библиографских записа на магнетну траку. Он не дефинише дужину или садржај појединачног записа, нити даје какво посебно значење било ком пољу, потпољу или индикатору. ISO 2709 само спецификује функцију преноса библиографског записа на магнетну траку, која се користи у имплементацији MARC библиографског формата у одређеним рачунарским системима. Садржај, дужину и структуру појединачних поља, потпоља и индикатора одређује сам MARC библиографски формат.

Сваки запис по ISO 2709 стандарду има три дела: лидер (*leader*), директоријум и променљиви део који садржи поља библиографског записа различитих дужина. У лидеру су дефинисани подаци који описују библиографски запис и омогућавају аутоматску обраду ISO записа. Дужина лидера је

From the physical aspect, the structure of MARC records may differ. Since the introduction of the MARC format, many different physical formats of MARC records have been defined. In this article, formats are presented chronologically. Since the introduction of XML (*eXtensible Markup Language*) [10] specification, the definition of the bibliographic record format has experienced a new dimension. XML is not only a new bibliographic record format: the appropriate specification of XML bibliographic record format allows easy processing of bibliographic records in bibliographic information systems. For implementation of functions, such as validation of records or their transformation to different structures, many ready tools have been developed in Java and most of them declared as open source.

2. ISO 2709 records

When the MARC format was created, computers were not as developed and as user-friendly as they are nowadays. Then the standards for data exchange, that enable the representation of data in simple, human readable formats, did not exist. Magnetic tape was the only physical media for data exchange between computers. Data on a magnetic tape was packed in sequential order and they had to be read in sequential order as well. Thus, every record on a magnetic tape began with a group of information describing the record content, such as: record length, record status, names of the fields in the record, etc. The ISO 2709 (*Format for information exchange*) [11] standard for exchange of MARC bibliographic and related records was created in accordance with the above mentioned requests.

The ISO 2709 standard specifies the method of transport of MARC bibliographic record fields on to the magnetic tape. It does not define the length or the content of individual records, nor does it assign any meaning to fields, subfields or indicators. ISO 2709 only specifies the function of bibliographic record transport on to the magnetic tape that is used for the implementation of MARC bibliographic format in certain computer systems. The content, length and structure of individual fields, subfields and indicators are determined by the MARC bibliographic format itself.

Each record according to the ISO 2709 standard has three parts: leader, directory and a data part, that contains bibliographic record fields of different lengths. The leader part defines parameters for bibliographic record description, enabling automatic processing of ISO records. It has 24 characters and

24 знака и садржи информацију о дужини записа, статусу записа, типу записа, библиографском нивоу, хијерархијском нивоу, дужини индикатора, дужини идентификатора поља и потпоља, потпуности записа, облику библиографског описа, кодној табели.

У директоријуму се наводе тагови поља из записа и позиције на којима почиње садржај одређеног поља. У променљивом делу наводи се садржај поља у редоследу навођења имена поља у директоријуму. За свако поље прво се исписује делимитер за поље и оба индикатора. Затим се за свако потпоље тога поља наводи делимитер за потпоље, идентификатор потпоља и садржај потпоља. За контролна поља исписују се само делимитери поља и садржај поља. На крају записа додаје се делимитер за запис. Навођењем тагова поља у директоријуму омогућено је, да се пре обраде целог записа одреде поља која су интересантна за обраду и она се могу обрађивати посебно.

Спецификацијом стандарда ISO 2709 дефинише се структура записа за размену података међу различитим рачунарским системима, а не за обраду података унутар одређеног рачунарског система. На ово се указује стога што је структура ISO записа сложена и непогодна за обраду података. Не само да је таква структура нечитљива за људе, већ захтева развој врло сложених програма за обраду садржаја у тим записима.

3. SGML записи

Развојем информационих технологија знатно су унапређени и рачунари и рачунарски системи. Поред могућности рачунара, боље су и могућности преноса података међу рачунарима и рачунарским системима. Развили су се нови физички носиоци података, као што су дискете, компакт дискови, DVD, итд. Данас су брзине физичких носилаца много пута веће него у време почетка њиховог развоја. Осим тога, претраживање података на њима није више секвенцијално, као што је било у време магнетне траке. У доба магнетне траке меморија је била врло скупа и програми су морали бити штедљиви код обраде података. Зато су се подаци обично организовали у линеарне, секвенцијалне структуре. Сада се подаци могу складиштити и у нелинеарним структурама и доступ до њих је свеједно брз и сигуран.

Осим развоја рачунара и физичких носилаца, на структуру података за пренос утицао је и развој Интернета. Сада се пренос података не врши само преко физичких медија, већ се највећа количина података преноси баш преко Интернета. Са развојем Интернета развио се и XML којим се покушава смањити хаос на Интернету.

contains the following information: record length, record status, type of record, bibliographic level, hierarchical level, indicator length, lengths of field and subfield identifiers, completeness of record, descriptive cataloguing form, character set table.

In the directory the tags of record fields and positions, on which their contents begin, are defined. The data part includes a variable number of entries, each corresponding to its respective field in order of its citation in the directory map. Each entry consists of field separator, both indicators, and of subfield separator, subfield identifier and subfield content, for each field's subfield. For control fields each entry includes only field separator and field content. Every record ends with record separator. Citation of field's tag in the directory map enables selection of fields interesting for individual processing, before processing of the whole record.

The ISO 2709 standard describes a generalized structure, a framework designated specially for communication between data processing systems and not for the use of a processing format within systems. This is emphasized on the grounds of the structure of ISO records being complex and inadequate for data processing. Not only it is human unreadable, but requires development of very complex programs for record content processing.

3. SGML records

The evolution of information technologies leads to the improvement of computer systems. In addition to improved computers, the abilities of data transfer between computers and computer systems have also become better. New media types have been developed, e.g. disquette, CD, DVD, etc. Today, their speeds are several times higher than they were at the beginning. Additionally, data searching is not sequential as it was at the time of magnetic tape. Then, the memory was very expensive and data processing programs had to be very economical, which resulted in data having to be organised in linear, sequential structures. Nowadays, however, data can be saved at non-linear structures, access to them is also fast and safe.

In addition to the development of computers and media types, Internet has a great influence on data structure as well. Today, data exchange is carried out over the Internet to a large extent and not so much by physical media. With the development of the Internet, the XML has also been developed and is one of the means for stopping the Internet chaos.

XML је једноставан, флексибилан, текстуални формат, развијен на основу стандарда SGML (*Standard Generalized Markup Language*) [12,13]. Развио га је W3C (*World Wide Web Consortium*) [14] са наменом да покрије широко подручје електронског издаваштва. Међутим, праву употребу XML је досегао на подручју размене најразличитијих података преко Интернета. SGML је развила међународна организација ISO (*International Organization for Standardization*) 1986. године под ознаком ISO 8879 на основу језика GML (*Generalized Markup Language*) [15] којег су крајем 60-тих година развили у фирми IBM.

XML и SGML су језици за означавање који омогућавају дефинисање нових формалних језика специфичних за одређене потребе. Ти језици садрже одређене XML/SGML елементе и атрибуте који се међусобно комбинују, што је одређено граматиком језика. Граматике тих нових језика дефинишу се са DTD (*Дефиниција типа документа*) [16] спецификацијом у којој се, посебним синтаксним правилима, описују елементи и атрибути језика, њихове могуће вредности и комбинације. Свака DTD спецификација дефинише посебну класу XML/SGML докумената.

MARC Standards Office [17] је средином 90-тих година развио две DTD спецификације за SGML документе, које подржавају конверзију каталошких података MARC записа без губљења података. У пројекту развијања ових DTD и алата за конверзије могли су учествовати сви заинтересовани корисници MARC-а и SGML-а. Тако је маја 1996. године настала прва *alpha* верзија ових DTD спецификација које су покривале пет MARC формата - *MARC формат за библиографске податке*, *MARC формат за информације заједнице*, *MARC формат за податке о стању фонда*, *MARC формат за нормативне податке* и *MARC формат за податке класификације*. Програми за конверзије MARC записа по ISO 2709 стандарду у MARC записе по SGML стандарду почели су се развијати 1997. године.

MARC DTD [18] спецификација дефинише граматiku MARC записа по SGML стандарду. Сви SGML тагови који одговарају пољима и потпољима, почињу са "mrcb" за записе *MARC формата за библиографске податке*, *MARC формата за информације заједнице*, *MARC формата за податке о стању фонда*, или са "mrca" за записе *MARC формата за нормативне податке* и *MARC формата за податке класификације*. Њима следе троцифрени идентификатор поља, нпр., "mrcb245" за MARC поље 245, или троцифрени идентификатор поља, цртица и ознака потпоља, нпр., "mrcb245-a"

XML is a simple, flexible, textual format based on SGML (*Standard Generalized Markup Language*) [12,13] standard. W3C (*World Wide Web Consortium*) [14] created it to cover a wide area of electronic publishing. But, the right use of XML is attained in the exchange of many different types of data over the Internet. SGML was developed in 1986 by ISO (*International Organization for Standardization*) under the name ISO 8879. It is based on GML (*Generalized Markup Language*) [15], a language created in the IBM at the end of the 1960s.

XML and SGL are markup languages for specifying new formal languages for certain requirements. Their grammar defines XML/SGML elements and attributes and their combinations. DTD (*Document Type Definition*) [16] specification defines their grammar with particular syntax rules for descriptions of elements and attributes of languages, their values and possible combinations. Every DTD specification defines one class of XML/SGML documents.

In the mid 1990s, the *Network Development and MARC Standards Office* [17] developed two SGML DTDs that supported the conversion of cataloging data from the MARC data structure to SGML (and back) without loss of data. The project of creating DTD specifications and conversion utilities was opened for input from any interested MARC and/or SGML users. The *alpha* version of the MARC DTDs was made available in May 1996; it handled five MARC formats: *MARC Format for Bibliographic Data*, *MARC Format for Community Information*, *MARC Format for Holdings Data*, *MARC Format for Authority Data*, *MARC Format for Classification Data*. In 1997, the work on the MARC-to-SGML conversion utilities began.

MARC DTD [18] specification defines the grammar of MARC record according to SGML standard. All SGML tags for fields and subfields in records of *MARC Format for Bibliographic Data*, *MARC Format for Community Information*, *MARC Format for Holdings Data* begin with "mrcb" and in records of *MARC Format for Authority Data*, *MARC Format for Classification Data* with "mrca". They are followed by a three-digit field identifier, for example "mrcb245" for MARC field 245, or by three-digit field identifier, hyphen (-), and the one-character subfield identifier, for example "mrcb245-a" for MARC

за MARC потпоље 245а. Имена SGML тагова за први и други индикатор поља су "i1" и "i2".

На сличан начин именују се SGML тагови за елементе који одговарају контролним пољима MARC записа. Уместо идентификатора потпоља у имену тага одговарајућег SGML елемента наведен је специјални код од два карактера и позиције одо до на којима се у MARC запису налази садржај описан тим SGML елементом. На пример, за позиције од 8 до 10 поља 006 назив SGML елемента је "mrcb006-SE-08-10", или за позицију 22 поља 008 назив SGML елемента је "mrcb008-BK-22".

Пошто је лидер у MARC запису фиксне дужине и свака позиција има своје значење, он нема придружени таг поља од три карактера, јединствен скуп елемената дат је дефиницијама MARC DTD за сваку могућу конфигурацију лидера. Као и са другим елементима података фиксне дужине, имена елемената лидера јесу ниске конкатенацијом настале од префикса "mrcaldr" или "mrcblldr", кода типа формата MARC ("bd", "hd", "ci", "ad", или "cl") којем претходи и следи цртица и број који идентификује позицију(е) карактера који се примењује(у). На пример, име SGML тага за позицију 5 у лидеру неког библиографског записа је "mrcblldr-bd-05".

Оваква DTD спецификација омогућава пренос и оних података у SGML записе из ISO записа, који нису потребни у SGML запису. На пример, информација о дужини ISO записа у лидеру је потпуно непотребна са становишта обраде SGML записа. Међутим, све информације сачуване су због лакше повратне конверзије из SGML записа у ISO запис. Из ISO записа у SGML запис нису пренети директоријум и делIMITЕРИ за поља, потпоља и записе, који се морају креирати приликом повратне конверзије.

4. XML записи

Са развојем XML-а је 2001. године DTD спецификација за SGML записе прерађена у DTD спецификацију за XML записе [19]. Њоме је дефинисана посебна класа XML докумената који представљају MARC записе. Посебну класу XML докумената могуће је дефинисати и XML шема језиком (*XML Schema Language*) [20, 21, 22]. XML шема је сте један од XML језика и користи се за писање дефиниција разних типова XML докумената. Предности XML шеме у односу на DTD су да XML шемом можемо дефинисати сложене типове података, просте типове података на основу већ уграђених простих типова података (на пример *date*, *string*, *boolean*, *positiveInteger*), више пута употребити једном дефинисане типове и слично.

subfield 245a. SGML tag names for first and second field indicators are "i1" and "i2".

SGML tag names for elements corresponding to the control fields in MARC records are defined in a similar way. Instead of subfield identifier in the SGML tag name there is special two-character code, followed with starting and ending positions, separated with a hyphen, that hold content of MARC record described by this SGML element. For example: for the positions from 8 to 10 in the field 006 the SGML tag name is "mrcb006-SE-08-10", or for the position 22 of the field 008 the SGML tag name is "mrcb008-BK-22".

Since the MARC record leader is a positionally-defined fixed-length data element and it does not have an associated three-character field tag, a unique element set was defined in the MARC DTDs for each possible leader configuration. As with other fixed-length data elements, the names of leader elements are concatenated strings consisting of: the prefix "mrcaldr" or "mrcblldr", a MARC format type code, preceded by a hyphen (one of the following: "bd", "hd", "ci", "ad" or "cl") and numeric identifier for the applicable character position(s), also preceded by a hyphen. For example: SGML tag name for position 5 in the leader of a bibliographic record is "mrcblldr-bd-05".

So-defined DTD specification allowed transmission of data from ISO to SGML records that are not necessarily in a SGML record. For instance, the information of ISO record length held in the leader part is entirely unnecessary from the position of SGML record processing. But, all information is saved for an easy reverse conversion from SGML to ISO records. Directory, as well as field, subfield and record separators, that must be created at reverse conversion, were not translated from ISO to SGML records.

4. XML records

In 2001, with XML specification development, the SGML DTD was fully converted to XML DTD [19], which determines particularly the class of XML documents representing MARC records. Also, the particular class of XML documents may be defined with XML schema (*XML Schema Language*) [20, 21, 22]. XML schema is one of the XML languages for defining different types of XML documents. The advantage of XML schema concerning DTD is that XML schema enables defining of complex data types, simple data types based on built-in simple data types (for example: *date*, *string*, *boolean*, *positiveInteger*), re-use of once-defined data types, etc.

Развојем XML шеме за MARC записе настао је MARC XML [23] формат за пренос MARC записа. То је XML формат, чија дефиниција у XML шеми одређује имена тагова за XML елементе који одговарају лидеру, пољима, потпољима и индикаторима MARC записа и њихове могуће вредности и комбинације. Пример записа у формату MARC XML дат је на **Листингу 1**.

```
<record>
<leader>01072nam a2200229 i 4500</leader>
<controlfield tag="001">564329</controlfield>
<controlfield tag="005">20010207</controlfield>
<controlfield tag="008">930423|1988||||xsub|||||||1||
||eng|c</controlfield>
<datafield ind1="" ind2="" tag="020">
<subfield code="a">0306429179</subfield>
</datafield>
<datafield ind1="" ind2="" tag="040">
<subfield code="a">XXYY</subfield>
<subfield code="b">srb</subfield>
</datafield>
<datafield ind1="0" ind2="" tag="041">
<subfield code="a">eng</subfield>
</datafield>
<datafield ind1="" ind2="" tag="080">
<subfield code="a">061.3</subfield>
</datafield>
<datafield ind1="1" ind2="" tag="111">
<subfield code="a">NATO Advanced Research
Workshop on the Structure of the Photosynthetic
Bacteria Reaction Center</subfield>
<subfield code="d">(1987 :</subfield>
<subfield code="c">Cadarache, France</subfield>
</datafield>
<datafield ind1="0" ind2="4" tag="245">
<subfield code="a">The photosynthetic bacterial
reaction center :</subfield>
<subfield code="b">structure and dynamics :
[proceedings of NATO Advanced Research
Workshop on the Structure of the Photosynthetic
Bacterial Reaction Centre...held September 20-25,
1987, Cadarache, France] /</subfield>
<subfield code="c">edited by Jacques Breton and
André Verméglio.</subfield>
</datafield>
```

Developing XML schema for MARC records has resulted in MARC XML [23] being created for work with MARC data in an XML environment. It is the XML format, in which the definition in XML schema determines tag names for XML elements corresponding to a leader, fields, subfields and indicators in MARC records and their possible values and combinations. **Example 1** shows a record in MARC XML format.

```
<datafield ind1="" ind2="" tag="260">
<subfield code="a">New York ;</subfield>
<subfield code="a">London :</subfield>
<subfield code="b">Plenum,</subfield>
<subfield code="c">1988</subfield>
</datafield>
<datafield ind1="" ind2="" tag="300">
<subfield code="a">X, 443 p. :</subfield>
<subfield code="b">graf. Pres. ;</subfield>
<subfield code="c">26 cm</subfield>
</datafield>
<datafield ind1="0" ind2="" tag="490">
<subfield code="a">NATO ASI series : advanced
science institutes series. Series A, Life
sciences,</subfield>
<subfield code="v">vol. 149</subfield>
</datafield>
<datafield ind1="0" ind2="4" tag="650">
<subfield code="a">Fotosinteza</subfield>
<subfield code="x">kongres</subfield>
</datafield>
<datafield ind1="0" ind2="4" tag="650">
<subfield code="a">Bakterije</subfield>
<subfield code="x">kongres</subfield>
</datafield>
<datafield ind1="1" ind2="2" tag="700">
<subfield code="a">Breton, Jacques.</subfield>
<subfield code="4">edt</subfield>
</datafield>
<datafield ind1="1" ind2="" tag="700">
<subfield code="a">Verméglio, André.</subfield>
<subfield code="4">edt</subfield>
</datafield>
</record>
```

Листинг 1: Запис у MARC XML формату
Example 1: Record in MARC XML format

Име тага XML елемента који одговара лидеру је "leader". Имена тагова XML елемената који одговарају контролним пољима су "controlfields". Ови XML елементи садрже атрибут "tag" са идентификатором поља. Имена тагова XML елемената који одговарају пољима су "datafields". Ови XML елементи садрже атрибут "tag" са идентификатором поља, и атрибуте "ind1" и "ind2" са вредностима првог и другог индикатора поља. Имена тагова XML елемената који одговарају потпољима су "subfields". Ови XML елементи садрже атрибут "code" са идентификатором потпоља.

The tag name of XML element corresponding to a leader is "leader". Tag names of XML elements corresponding to control fields are "controlfields". These XML elements have an attribute "tag" with field identifier. Tag names of XML elements corresponding to a fields are "datafields". These XML elements have the attribute "tag", with field identifier, and the attributes "ind1" and "ind2", with values of the first and the second indicator. Tag names of XML elements corresponding to subfields are "subfields". These XML elements have the attribute "code" with subfield identifier.

У MARC XML запис пренети су сви битни подаци из MARC ISO записа. Међутим, за разлику од XML записа који је дефинисан са MARC DTD, у MARC XML запис нису пренети неки подаци везани за структуру MARC ISO записа. На пример, лидер је пренет у један елемент "leader", а не у више елемената који зависе од позиције, као што је, на пример, елемент "mrcbldr-bd-05" у MARC DTD запису. Слично је и са контролним пољима. Овај губитак података није од важности за повратну конверзију записа у формат ISO 2709, јер се све ове информације могу одредити из садржаја MARC XML записа.

Предности овакве структуре MARC XML записа су вишеструке:

- пренос оваквих записа у друге мета формате је једноставан, на пример, са XSLT (*XSL Transformation*) [24] трансформацијама,
- записи у овом формату су читљивији за човека,
- обрадом оваквих записа је на лакши начин могуће доћи до аналитичких података о записима и извршити валидацију тих записа,
- програме за рад са записима у овом формату је много лакше развијати, него програме за рад са записима у формату ISO 2709.

Међутим, недостатак MARC XML формата је неприлагођеност корисницима. Овако дефинисани MARC XML записи садрже и информације које, са становишта обраде XML записа, нису потребне, као на пример, XML елемент "leader". Садржај у неким XML елементима, једнак је садржају у MARC ISO записима и стога нечитљив за корисника, као на пример, у XML елементима "leader" и "controlfield". Осим тога, имена тагова свих XML елемената који одговарају пољима из променљивог дела су једнака - означена су као "datafields", што смањује прегледност записа у оваквом формату.

Како је овај формат више намењен преносу и XSLT трансформацији XML библиографских записа у библиотечким системима, а не употреби од стране корисника, у Конгресној библиотеци су 2002. године развили нов XML формат библиографских записа - MODS (*Metadata Object Description Schema*) [25]. MODS формат садржи изабран подскуп MARC поља и више је прилагођен корисницима него што је то случај са форматом MARC XML. Пример једног MODS записа дат је на **Листингу 2**.

MODS има своју формалну дефиницију у XML шеми која је, за разлику од XML шеме за MARC XML записе, разумљивија и прилагођена корисницима. Наиме, имена тагова XML елемената у MODS запису су енглеске речи, чије значење

MARC XML records contain all essential data from MARC ISO records. Some informations on structure of MARC ISO records, however, do not transmit to MARC XML record, as in the case of XML record defined with MARC DTD. For example, leader is transmitted to one MARC XML element "leader" and not to several positionally defined XML elements, e.g. XML element "mrcbldr-bd-05" in MARC DTD record. Control fields behave in a similar way. This loss of information is not of importance for reverse record conversion to ISO 2709 format, because all information can be determined from the content of MARC XML record.

Advantages of the structure of MARC XML records defined in this way are numerous. They are as follows:

- transformation of these records in some other meta formats is simple, for instance with XSLT (*XSL Transformation*) [24],
- records in this format are human readable,
- processing of records in this format produces analytical output and validation in a simple way,
- programs for record processing in this format are not as complicated as programs for working with records in ISO 2709 format.

The disadvantage of MARC XML format is that it is not end user oriented. MARC XML records defined in this way hold unnecessary information on XML record processing, too, in the XML element "leader". The content of some XML elements, e.g. "leader" or "controlfield", equals to the content of MARC ISO record. Therefore, it is unreadable for users. Moreover, tag names of all XML elements, corresponding to fields from the variable part, are equal, they are referred to as "datafields". This results in a reduced clarity of records in this format.

This format is more applicable for data exchange and XSLT transformations of XML bibliographic records in library systems. In 2002, the Library of Congress developed a schema for a bibliographic element set that could be used for a variety of purposes and defined the new XML format of bibliographic records, referred to as MODS (*Metadata Object Description Schema*) [25]. It includes a subset of MARC fields and is more user oriented than MARC XML format. **Example 2** shows a record in MODS format.

MODS has its formal definition in XML schema that is more understandable and user-friendly than XML schema for MARC XML records. MODS format uses language-based tags, the meanings of which correspond to the contents of fields and

одговара садржају поља и потпоља библиографског записа, док се у MARC XML формату XML елементи који одговарају пољима разликују само по бројевима у њиховим атрибутима. На пример, име тага MODS елемента који одговара наслову из поља 245 је "title", име тага MARC XML елемента је "subfield", а тачно одређење наслова дато је атрибутом "tag=245" овог XML елемента. У неким случајевима се одређени XML елементи из MARC XML формата групишу у један XML елемент MODS формата. У примерима, на **Листингу 1 и 2** који представљају исти библиографски запис у форматима MARC XML и MODS, неки подаци из MARC XML елемента "controlfield" који одговара пољу 008 библиографског записа, и подаци из MARC XML елемента "subfield" који одговарају потпољима поља 206 библиографског записа, стапају се у XML поделемента MODS елемента "publicationInfo".

```
<mods>
<titleInfo>
  <nonSort>The</nonSort>
  <title>photosynthetic bacterial reaction center :
    structure and dynamics : [proceedings of NATO
    Advanced Research Workshop on the Structure of
    the Photosynthetic Bacterial Reaction Centre...held
    September 20-25, 1987, Cadarache, France] /</title>
</titleInfo>
<name type="conference">
  <namePart>NATO Advanced Research Workshop on
  the Structure of the Photosynthetic Bacteria Reaction
  Center (1987 : Cadarache, France)</namePart>
  <role>creator</role>
</name>
<name type="personal">
  <namePart>Breton, Jacques.</namePart>
  <role>edt</role>
</name>
<name type="personal">
  <namePart>Verméglio, André.</namePart>
  <role>edt</role>
</name>
<typeOfResource>text</typeOfResource>
<genre>conference publication</genre>
<publicationInfo>
  <placeCode authority="marc">xxu</placeCode>
  <place>New York ;</place>
  <place>London :</place>
  <publisher>Plenum,</publisher>
  <dateIssued>1988</dateIssued>
  <issuance>monographic</issuance>
</publicationInfo>
```

subfields, rather than numerical ones, used in MARC XML format. For example, the tag name of MODS element, corresponding to the title in the field 245, is "title", but the tag name of MARC XML element is "subfield" and the exact determination of title is given in the attribute "tag=245" of this XML element. In some cases MODS format regrouping elements from MARC XML format and sometimes several data elements in MARC XML format may be brought together into one in MODS element. In **Examples 1 and 2**, showing the same bibliographic record in MARC XML and MODS formats, some information forms the MARC XML element "controlfield", corresponding to the control field 008 in a bibliographic record, and some information forms the MARC XML element "subfield", corresponding to the field 206 in a bibliographic record, are merged in XML subelements of the MODS element "publicationInfo".

```
<language authority="iso639-2b">eng</language>
<physicalDescription>
  <extent>X, 443 p. : graf. pres. ; 26 cm</extent>
</physicalDescription>
<note type="statement of responsibility">edited by
  Jacques Breton and André Verméglio.</note>
<subject>
  <topic>Fotosinteza</topic>
  <topic>kongres</topic>
</subject>
<subject>
  <topic>Bakterije</topic>
  <topic>kongres</topic>
</subject>
<classification authority="udc">061.3</classification>
<relatedItem type="series">
  <titleInfo>
    <title>NATO ASI series : advanced science institutes
    series. Series A, Life sciences, vol. 149</title>
  </titleInfo>
</relatedItem>
<identifier type="isbn">0306429179</identifier>
<recordInfo>
  <recordContentSource>XXYY</recordContentSource>
  <recordCreationDate
    encoding="marc">930423</recordCreationDate>
  <recordChangeDate
    encoding="iso8601">20010207</recordChangeDate>
</recordInfo>
</mods>
```

Листинг 2: Запис у MODS формату **Example 2: Record in MODS format**

Скуп XML елемената је у MODS формату мањи од скупа XML елемената у MARC XML формату који се показао сувише опширним за потребе преноса библиографских података у протоколу Z39.50 [26]. У овом протоколу за сада још увек је највише заступљен формат ISO 2709.

The element set of MODS format is simpler than the one of MARC XML format that is to be comprehensive for data exchange in Z39.50 protocol [26]. For the time being, ISO 2709 format is the most considered by this protocol. Since the introduction of XML, it has become data exchange format in this

Од појаве XML-а подаци су почели да се преносе и у XML формату, и то најчешће у формату DC (од *Dublin Core - Даблинско језгро*) [27]. Међутим, DC формат, који садржи 15 основних елемената, употребљив је више у размени података између библиотечких система и издавача, него што је то случај у размени између разних библиотечких система. Скуп елемената из формата MODS компатибилнији је са библиотечким подацима од формата ONIX [28] који је такође и формат за размену података у протоколу Z39.50. С обзиром на карактеристике свих ових формата, MODS формат који садржи оптималну меру података, реалан је кандидат за формат Z39.50 Next Generation.

5. Валидација XML записа

Једна од највећих предности XML записа у односу на ISO записе је могућност лакше обраде. XML стандард је релативно нов. Међутим, за рад са XML документима већ су развијени јако добри алати. Највише њих израђено је у програмском језику Јава и декларисано као отворени код. Основни програми за рад са XML документима део су JDK 1.4 [29]. Како је сваки XML библиографски запис једна инстанца XML документа, за обраду XML записа могу се употребити сви ти алати и не морају се улагати средства у развој сопствених програма, као што је то био случај код ISO записа.

XML записи имају своју дефиницију у XML шеми, што омогућава њихову валидацију у односу на правила одређена том XML шемом. На пример, XML шемом за MARC XML записе одређени су елементи који могу наступати у MARC XML записима - "leader", "controlfield" и "datafield", и њихов начин гњеждења. XML шема за MARC XML записе је јавна и може свако да је пренесе са Интерета у свој систем и валидира своје MARC XML записе.

За валидацију XML докумената по XML шеми постоје већ развијени програми – валидатори, који су углавном декларисани као отворени код. То омогућава да се без трошкова набавке, на једноставан начин, уграде у постојеће програме локалних библиотечких система. Валидатори се међусобно разликују по функционалности у односу на то по којим шемама валидирају XML документе (W3C XML Schema Language, Schematron [30], RELAX NG [31], TREX [32], SOX [33], XDR [34], HOOK [35], DSD [36], Assertion Grammars [37], xlinkit [38] итд.), да ли омогућавају комбинације елемената више различитих врста шема у документу једне XML шеме, да ли омогућавају складиштење шема, односно омогућавају да се једном учитане шеме употребе за валидацију више

protocol and data is mostly in DC (*Dublin Core*) format [27]. DC format, however, consists of a set of 15 basic elements and is much more usable for data exchange between library systems and publishers than between different library systems. The element set of MODS format is more compatible with library data than ONIX [28] format that is also the format for data exchange in Z39.50 protocol. Regarding the characteristics of all these formats, MODS contains the most optimal data and it is a very interesting candidate for the Z39.50 Next Generation specified format.

5. Validation of XML records

Easy data processing is one of the most significant advantages of XML records if compared to ISO records. Even though XML standard is relatively new, many good tools for work with XML documents have been already developed. Most of them are in Java programming language and declared as open source. The basic programs for work with XML documents are a part of JDK 1.4 [29]. Each XML bibliographic record is an XML document, for the processing of which all of the XML tools may be used. Therefore, it is not necessary to invest in new program development, which was required with ISO records.

The definition of XML records is in XML schema. This enables validation of XML records according to the rules specified in XML schema. For example, XML schema of MARC XML records determines XML elements that can be included and nested in MARC XML records: "leader", "controlfield" and "datafield". This XML schema for MARC XML records is public and everyone can download it from the Internet and perform validation of its MARC XML records.

For XML documents validation, ready tools, validators, exist and they are mostly declared as open source. This enables a simple way to build them in existing programs of local systems, without acquisition outlay. Validators differ from each other in view of their functionality, i.e. on which schemas they validate XML documents (W3C XML Schema Language, Schematron [30], RELAX NG [31], TREX [32], SOX [33], XDR [34], HOOK [35], DSD [36], Assertion Grammars [37], xlinkit [38], etc.), if they allow combinations of elements of different schema types in one schema document, if they enable schema caching – to use the same saved schema in validation of several XML documents, if they are *fail-fast* - report a validation error in the beginning or at the end of an XML element, if they report all validation errors

XML докумената, да ли су *fail-fast*, односно да ли јављају поруку о грешци на почетку или на крају XML елемента, да ли јављају све грешке одједном или само прву, и да ли у потпуности имплементирају одређен шема језик.

Један од бољих валидатаора је MSV (*Sun Multi-Schema XML Validatator*) [39] који имплементира интерфејс JARV [40] за валидацију XML докумената. MSV омогућава валидацију XML докумената према следећим шема језицима: RELAX NG, TREX, W3C XML Schema и XML DTD. Осим тога, омогућава складиштење шема, што увелико убрзава валидацију XML докумената, јер се шема парсира само једном код читавања у меморију. MSV је и *fail-fast* и јавља све грешке одједном, за разлику од, на пример, XMLSpy едитора [41] који јавља само прву грешку, што је слаба страна валидатаора са становишта поправљања грешака у XML документима. XMLSpy за разлику од MSV валидатаора не имплементира W3C XML Schema језик у потпуности, што може довести до неисправних резултата валидације XML докумената.

На брзину валидације XML докумената утиче и XML парсер којим се чита XML документ. У MSV се може укључити било који XML парсер који имплементира интерфејс JAXP [42] са SAX (*Simple API for XML*) [43] и DOM (*Document Object Model*) [44,45,46,47,48,49,50] парсерима. Део JDK 1.4 је и парсер Xerces [51] који се најчешће употребљава за парсирање XML докумената. Он укључује и валидатор који има слабије могућности од MSV валидатаора и примернији је за валидацију једноставнијих XML докумената. На пример, Xerces омогућава валидацију само по W3C XML Schema језику и не омогућава складиштење шеме.

Валидацијом XML записа могу се селектovati записи који не одговарају стандардима пре преношења у локални библиотечки систем. Наиме, валидацијом по одређеној XML шеми добијају се поруке о грешкама у XML запису у односу на правила дефинисана у тој XML шеми. Тим правилима дефинишу се обавезност и поновљивост XML елемената, имена, типови и обавезност њихових атрибута, типови података у XML елементима и све могуће комбинације тих XML елемената. Тако је, на пример, шемом MARC XML записа дефинисано да су називи XML елемената који одговарају свим пољима променљивог дела исти - "datafield", а тачније одређивање идентификатора поља је у атрибуту "tag" (**Листинг 1**). Како се у XML шеми елемент "datafield" може дефинисати само једанпут, његова обавезност и поновљивост у XML запису једнозначно је одређена за сва појављивања тог XML елемента. Значи да се по тој XML шеми не могу проверити

at once or only the first error, and if they fully implement certain schema language.

MSV (*Sun Multi-Schema XML Validatator*) [39] is one of better validators. MSV implements JARV [40] interface for validation of XML documents according to the following schema languages: RELAX NG, TREX, W3C XML Schema and XML DTD. Additionally, MSV enables schema cacheing that makes validation of XML documents quicker, as schema is parsing into the memory only once. MSV is fail-fast and reports all validation errors at once, unlike for example, XMLSpy editor [41] that reports only the first validation error. This is a disadvantage from the position of error corrections in XML documents. XMLSpy unlike MSV does not fully implement the W3C XML Schema language. This may lead to irregular XML validation results.

The speed of XML document validation depends on XML parser that is used for reading XML documents. MVS may incorporate any XML parser that implements JAXP interface [42] with SAX (*Simple API for XML*) [43] and DOM (*Document Object Model*) [44,45,46,47,48,49,50] parsers. XML parser Xerces [51] is most widely used; it is a part of JDK 1.4. It includes the XML validator that is most suitable for validation of simpler XML documents, because it does not have such a wide range of possibilities as MSV. For example, Xerces allows validation only according to the W3C XML Schema language and does not allow schema cacheing.

Validation of XML documents enables selecting poor quality records before saving them to a local library system. This results from the fact that validation according to the chosen XML schema gives information on deviations in XML records concerning the rules defined in this XML schema. The rules define obligatoriness and repeatability of XML elements, names, types and obligatoriness of their attributes, data types in XML elements and all combinations of XML elements possible. For example, the schema of MARC XML records defines the same name for XML elements, i.e. "datafields", corresponding to all fields in the variable part of bibliographic records. The attribute "tag" in XML (**Example 1**) is a precise determination of field identifier. In XML schema, the element "datafield" can be defined only once, therefore, the obligatoriness and repeatability are uniformly defined for all appearances of the element "datafield" in an XML record. Thus, XML schema cannot verify obligatoriness and

обавезност и поновљивост XML елемената "datafield" који одговарају различитим пољима библиографског записа. На пример, у XML шеми MARC XML записа није могуће дефинисати поновљивост елемента "datafield" са атрибутом "tag=700", који одговарају пољима 700 са подацима о ауторима, и непоновљивост елемента "datafield" са атрибутом "tag=245", који одговара пољу 245 са подацима о наслову.

Слично се XML шемом MARC XML записа не могу проверавати ни обавезност и поновљивост XML елемената који одговарају потпољима у пољима библиографских записа. Оваквом структуром XML записа не може се проверити чак ни да ли је неко поље или потпоље, дефинисано одређеним XML елементом у XML запису, уопште дефинисано MARC форматом.

Да би се XML шемом могле проверавати ове особине, називи XML елемената морали би бити различити за свако различито поље, односно потпоље унутар поља библиографског записа. На пример, XML елемент који одговара пољу 650 библиографског записа могао би изгледати као на следећем примеру:

```
<field650>
  <ind1>0</ind1>
  <ind2>4</ind2>
  <subfielda>Bakterije</subfielda>
  <subfieldx>kongres</subfieldx>
</field650>
```

По овом формату је свако поље, потпоље и индикатор представљено својим XML елементом без атрибута. Валидацијом овакве структуре могу се добити поруке о више грешака него валидацијом MARC XML структуре. Исто тако, валидацијом MODS записа могу се добити значајније информације о грешкама у библиографском запису него валидацијом MARC XML записа, с обзиром на структуру MODS записа код које су имена XML елемената различита и одговарају садржају поља односно потпоља.

6. Трансформација XML записа

Валидацијом XML докумената проверавају се XML елементи у односу на правила дефинисана у тој XML шеми, као што су, на пример, постојање, обавезност или поновљивост XML елемената. Међутим, XML шемом не могу се проверити садржаји свих XML елемената или међусобна зависност XML елемената. Због тога се валидацијом XML записа не могу проверити сви концепти садржаја и зависности поља и потпоља у библиографском запису. На пример, XML шемом не може се проверити да ли садржај одређеног поља стварно представља ISBN број (да ли облик

repeatability of the XML element "datafield", corresponding to different fields in a bibliographic record. For example, schema for MARC XML records does not make it possible to define repetition of the XML element "datafield" with the attribute "tag=700" that corresponds to the field 700 with author information, and to define nonrepetition of the XML element "datafield" with the attribute "tag=245" that corresponds to the field 247 with title information.

By analogy, XML schema cannot verify obligatoriness and repeatability of the XML element "subfield" corresponding to the subfields in bibliographic record fields. This structure of XML records cannot verify even, if some field or subfields, that are defined with particular XML elements in an XML record, exist in MARC format.

The tag name of every XML element, corresponding to each field or subfield in a bibliographic record, should be different to make it possible for XML schema to verify such characteristics of bibliographic records. For example, XML element, corresponding to field 650, may read as:

```
<field650>
  <ind1>0</ind1>
  <ind2>4</ind2>
  <subfielda>Bacteria</subfielda>
  <subfieldx>congress</subfieldx>
</field650>
```

According to this format, each field, subfield and indicator are represented with their own XML element, without attributes. Validation of the structure defined in this way obtains more information on errors in a bibliographic record than validation of the MARC XML structure. Also, by validation of MODS records more error messages are obtained than by validation of MARC XML records, because MODS record structure defines that the tag name of each XML element, corresponding to the bibliographic field or subfield, is different and depends on field/subfield content.

6. Transformation of XML records

XML document validation verifies XML elements included according to the rules specified in XML schema, such as existence, obligatoriness and repeatability of XML elements. In spite of that, XML schema cannot verify the content of all XML elements and their dependency. Thus, validation of XML records with XML schema cannot verify all concepts of the contents of subfields and fields and their dependencies in bibliographic records. For example, XML schema does not verify if some subfield content represents an ISBN number (if its form and control number conforms to the relevant standard), or if the

и контролни број одговарају стандарду), или да ли је година издавања у потпољу 100d већа или једнака години издавања у потпољу 100c. Да би се проверила сва оваква ограничења нужно је допунити XML шеме другим алатима [52].

Једна од првих могућности јесте описом додатних ограничења коришћењем једног или више других шема језика. На пример, додатна ограничења из Schematron језика уграђују се у елементе *appinfo* у документ XML шеме, као у следећем примеру:

```
<pattern>
    <rule>
        <assert>...</assert>
        <assert>...</assert>
    </rule>
</pattern>
```

За препознавање структуре Schematron језика потребна је Schematron *машина* (Schematron *engine*). Предност овог приступа је да су сва ограничења моделирају у једном а не у више докумената. Слаба страна је та што сваки језик шеме има одређене могућности и ограничења. На пример, Schematron језик не подржава променљиве и петље. Стога би за моделирање свих додатних ограничења било потребно више шема језика, а тиме и више времена за проучавање тих шема језика.

Други начин контролисања библиографских записа био би развој програма за проверу додатних ограничења у програмским језицима, као што су Java, Perl, C++ и слично. Предност овог приступа је да се овим програмским језицима могу моделирати све контроле, без обзира на њихову сложеност. Слаба страна је што је програме потребно компилirati код сваке њихове промене или допуне, што није потребно код шема докумената.

Трећи начин би било писање стилског листа за проверу ограничења у језицима XSLT и XPath (*XML Path Language*) [53] који су део стандарда XSL (*eXtensible Stylesheet Language*) [54]. XSLT омогућава претварање једног XML документа у други XML документ или у један од формата за представљање садржаја, као што је HTML, PDF, RTF, или у текст. При томе је омогућено сортирање, измена структуре и филтрирање XML елемената и њихових атрибута. Скуп таквих трансформација назива се *стил*. За једну класу XML докумената може се дефинисати више стилова, што омогућава претварање једног XML документа у више различитих формата.

year of publication in the subfield 100d is greater than or equal to the year of publication in the subfield 100c. To check all these constraints it will be necessary to supplement XML Schemas with another tool [52].

The first option is to use one (or more) of other schema languages to express additional constraints. For example, additional constraints of Schematron language are embedded in *appinfo* elements within the XML Schema document, e.g.:

The Schematron engine is necessary for recognition of Schematron structure. The advantage of this approach is that all of the constraints are expressed within one document rather than being dispersed over multiple documents. The disadvantage is that each schema language has its own capabilities and limitations. For example, Schematron language does not support loops and variables. Therefore, multiple schema languages may be required to express all the additional constraints and so, much more time is needed to learn each of the schema languages.

The second option is to write some Java, Perl, C++, etc. codes to check additional constraints. The advantage of this option is that with a single programming language you can express all the additional constraints, regardless of their complexity. The disadvantage is that programs must go through compiling, linking and executing, which is not required with schema documents.

The third option is to write a stylesheet to check the constraints with XSLT and XPath (*XML Path Language*) [53] languages which are parts of XSL (*eXtensible Stylesheet Language*) [54] standard. XSLT is a language for transforming XML documents into other XML documents or in some format for content representation, e.g. HTML, PDF or RTF, or in a text format. Thereby, XML elements and their attributes may be sorted, filtered and reordered. A transformation expressed in XSLT is called a stylesheet. For one class of XML documents many stylesheets may be defined for transforming one XML document into many different formats.

Предност овог приступа је у томе што је стилски лист са описима трансформација XML записа независан од XML шеме. XSLT и XPath су основне XML спецификације, већ имплементирани (на пример, XSLT трансформатор Xalan [55] је део JDK 1.4) и лако разумљиве, са доста документације. Језици XSLT и XPath су "богати" и њима се могу описати готово сва ограничења без учења других језика шема. Једина слабост овог приступа је у томе што се мора управљати два датотекама: документ XML шеме и документ XSLT/XPath за опис додатних ограничења морају се написати засебно. Неопходно је пажљиво управљање овим документима и чување за случај потребе.

У пракси се најчешће користи ова трећа опција. Значи, прво се за проверу што већег броја ограничења исцрпе све могућности XML шема, затим се за сва друга ограничења пише стилски лист за проверу и коришћење неког програмског језика ради контроле евентуално преосталих непроверених ограничења.

Валидацијом MARC XML записа XML шемом не може се проверити доста особина MARC записа, због једноставности структуре MARC XML записа. Како је MARC XML запис валидан XML документ, за контролисања преосталих ограничења MARC записа може се дефинисати XSLT спецификација са одговарајућим описима трансформација XML записа. Тако је Конгресна библиотека дефинисала XSLT спецификацију за валидацију MARC XML записа [56].

XSLT спецификација може се дефинисати и за конвертовање XML записа из једног у други XML формат. На пример, ако записи у локалном библиотечком систему нису MARC XML записи, већ записи у неком другом XML формату, морају се код размене записа конвертовати у одговарајући локални XML формат. За конверзију MARC XML записа у формате MARC DTD, MODS, RDF са DC и HTML, и за конверзију записа из формата MODS, DC, OAI, ONIX и MARC DTD у MARC XML формат, Конгресна библиотека дефинисала је посебне XSLT стилове [57]. На жалост, XSLT трансформацијама не могу се конвертовати MARC XML записи у ISO 2709 формат. За ту конверзију може се употребити Java пакет marc4j [58] који је декларисан као отворени код. У њему је дефинисан објекат *Record* који представља MARC запис и садржи метод *marshal* за конверзију учитаног MARC XML записа у формат ISO 2709.

The advantage of this option is that a stylesheet with descriptions of XML transformation is independent from XML schema. XSLT/XPath is a "core technology" which is well supported (for example, XSLT transformator Xalan [55] is a part of JDK 1.4), well understood, and with lots of material written on it. XSLT/XPath is a very powerful language. Most constraints (if not all of them) that might be ever needed can be expressed using XSLT/XPath. Thus, it is not necessary to learn multiple schema languages to express these additional constraints. With this approach, XML Schema document and XSLT/XPath document to express additional constraints must be written in separate documents. It is necessary to carefully manage the two documents and to keep them in synch needs.

In practice this third option is used to the greatest extent. Thus, the methodology for verifying constraints is as follows: check as many constraints as possible using XML schemas, for all other constraints write a stylesheet to do the checking and use some programming language to check the rest of constraints, if some of them are left unchecked.

Validation of MARC XML records with XML schema does not verify many constraints of MARC records, because the structure of MARC XML records is too simple. Since a MARC XML record is a valid XML document, it may be possible to define XSLT specification with adequate transformations for checking additional constraints of MARC records. Thus, the Library of Congress created XSLT for validation of MARC XML records [56].

XSLT specification may be defined for converting XML records from one to another XML format. For example, if records in local library system are not in MARC XML format, but in some other XML format, they should be converted into a certain local XML format after exchange. The Library of Congress created separate stylesheets [57] for conversions of MARC XML records into MARC DTD, MODS, RDF with DC or HTML formats, and for conversions of records from MODS, DC, OAI, ONIX or MARC DTD formats into MARC XML format. MARC XML records, however, cannot be converted into ISO 2709 format with XSLT transformations. Java package marc4j [58], that is declared as open source, may be used for this conversion. In this package the object *Record* is defined for representing MARC record structure and includes the *marshal* method for converting MARC XML record into ISO 2709 format.

7. Закључак

Од настанка библиографског формата MARC технологија преноса података међу рачунарима и међу различитим рачунарским системима јако се развила, што је допринело развоју нових стандарда на том подручју. Развој MARC записа ишао је од формата прилагођеног магнетној траци - ISO 2709, до XML формата који је прихватљивији за корисника, као и за стручњаке који се баве обрадом тих записа. Развој алата за обраду MARC записа у XML формату није више препуштен само библиографским сервисима, као што је то био случај са ISO форматом, већ у категорији програма отвореног кода постоји велики број решења која се на лак начин могу уградити у постојеће библиотечке информационе системе. XML формат је прихватљивији и за нове протоколе преноса података преко Интернета и Web сервиса [59]. Web сервиси су последњих година предмет великог интересовања у Интернет технологији. У неким областима их можемо већ користити. Конгресна библиотека такође развија своје Web сервисе.

Упркос развоју нових формата, ISO формат MARC записа остаје у употреби због огромног броја библиотечких и других система који користе библиографске податке и који још увек употребљавају старе рачунарске системе и технологије. Неки од тих система сигурно неће још друго трајати, док се неки надограђују у циљу имплементације нових технологија и стандарда, као што су Java и XML, и тиме обезбеђују себи постојање и у будућности. Вероватно ће протећи још много времена док сви системи не пређу на нову технологију, а до тада MARC запис остаје у свим својим форматима.

References

- [1] *The Library of Congress*. <http://www.loc.gov/>
- [2] *MARC Format Overview*. Network Development and MARC Standards Office, Library of Congress, 08/13/2003. <http://www.loc.gov/marc/status.html>
- [3] *MARC Standards*. Network Development and MARC Standards Office, Library of Congress, 09/16/2003. <http://lcweb.loc.gov/marc/>
- [4] *The UKMARC Exchange Record Format*. The British Library, National Bibliographic Service. <http://www.bl.uk/services/bibliographic/marc/marcexchange.html>
- [5] *UNIMARC Manual : Bibliographic Format 1994*. International Federation of Library Associations and Institutions. <http://www.ifla.org/VI/3/p1996-1/sec-uni.htm>
- [6] *UNIMARC Manual : Bibliographic Format 1994, APPENDIX H: Cataloguing Rules and Formats Codes*. International Federation of Library Associations and Institutions. <http://www.ifla.org/VI/3/p1996-1/appx-h.htm>
- [7] *Family of ISBDs : Publication list*. International Federation of Library Associations and Institutions, 2003. <http://www.ifla.org/VI/3/nd1/isbdlist.htm>
- [8] *Anglo-American Cataloguing Rules*. Amer Library Assn Editions, 2002.

7. Conclusion

Since the introduction of MARC bibliographic format, technology of data exchange between computers and between different computer systems has reached a very sophisticated stage and has contributed to the creation of new standards in this field. MARC records have been evolved from the format conformed to magnetic tape, i.e. ISO 2709, to the more user-friendly format that is acceptable also for record processing experts. Developing of tools for MARC record processing in XML format is not left only to bibliographic services as was the case with ISO 2709 records. Nowadays, in the category of open source many tools exist and they can be easily built-in into bibliographic information systems. XML is also a suitable format for data transfer on the Internet and for Web Services [59]. In recent years, Web Services have been the most interesting subject in the Internet technology. In some fields they already exist. The Library of Congress is also developing their Web Services.

Although new formats are being developed, ISO format of MARC records is still in use due to the existence of many library and other systems, using this data format and old computer systems and technologies. Some of these systems will cease to exist in the future. Some of them are expanding by the introduction and implementation of new technologies and standards, e.g. of Java and XML, to assure their existence in the future. Probably many years will pass before all systems transfer to new technologies. Up to then, MARC record will be present in all its formats.

- [9] *Library of Congress Subject Headings - Principles of Structure and Policies for Application: Contents*. Library of Congress, 1999. <http://www.tlcdelivers.com/tlc/crs/shed0014.htm>
- [10] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, editors. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation, 2000. <http://www.w3.org/TR/REC-xml>
- [11] *Information and documentation – Format for information exchange*. International Organization for Standardization, 1996.
- [12] *International Standard ISO 8879 Information Processing - Text and Office Systems -Standardized Generalized Markup Language (SGML), First Edition*. International Organization for Standardization, 1986.
- [13] *Standardized Generalized Markup Language (SGML): SGML and XML as (Meta-) Markup Languages*. Oasis, 2002. <http://xml.coverpages.org/sgml.html>
- [14] *World Wide Web Consortium (W3C)*. MIT, ERCIM, Keio, 1994-2003. <http://www.w3.org/>
- [15] C. Goldfarb. *A Brief History of the Development of SGML*. SGML Users' Group, 1990. <ftp://ftp.ifi.uio.no/pub/SGML/Newsire/027.1993-07-21>
- [16] *W3C XML Specification DTD ("XMLspec")*. ArborText Inc., 1999. <http://xml.coverpages.org/xmlspec-report-19990205.html>
- [17] *Network Development and MARC Standards Office*. Library of Congress, 02/19/02. <http://www.loc.gov/marc/ndmso.html>
- [18] *MARC SGML Archive*. Network Development and MARC Standards Office, Library of Congress, 05/22/2003. <http://www.loc.gov/marc/marcsgmlarchive.html>
- [19] *MARC XML DTD (Data Type Definition)*. Library of Congress, 01/16/2003. <http://www.loc.gov/marc/marcxml.html#marcdtd>
- [20] D. C. Fallside, editor. *XML Schema Part 0: Primer*. W3C Recommendation, 2001. <http://www.w3.org/TR/xmlschema-0/>
- [21] H. S. Tomphson, D. Beech, M. Maloney, N. Mendelsohn, editors. *XML Schema Part 1: Structures*. W3C Recommendation, 2001. <http://www.w3.org/TR/xmlschema-1/>
- [22] P. V. Brion, A. Malhotra, editors. *XML Schema Part 2: Datatypes*. W3C Recommendation, 2001. <http://www.w3.org/TR/xmlschema-2/>
- [23] *MARC 21 XML Schema*. Library of Congress. <http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd>
- [24] J. Clark, editor. *XSL Transformations (XSLT), Version 1.0*. W3C Recommendation, 1999. <http://www.w3c.org/TR/xslt>
- [25] *Metadata Object Description Schema (MODS)*. Library of Congress, 11/25/2003. <http://www.loc.gov/standards/mods/>
- [26] *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification*. National Information Standards Organization, 2002. <http://www.niso.org/standards/resources/Z39-50-200x.pdf>
- [27] *Dublin Core Metadata Initiative (DC)*. DCMI, 1995-2003. <http://dublincore.org/>
- [28] *ONIX for Books*. EDItEUR. <http://www.editeur.org/onix.html>
- [29] *Java 2 Platform, Standard Edition (J2SE) v 1.4*. Sun Microsystems, Inc., 1995-2003. <http://java.sun.com/j2se/1.4/>
- [30] R. Jelliffe. *The Schematron : An XML Structure Validation Language using Patterns in Trees*. Academia Sinica Computing Centre, Taipei, 1999-2001. <http://www.asc.net/xml/resource/schematron/schematron.html#overview>
- [31] J. Clark, M. Murata, editors. *RELAX NG Specification*. OASIS, 2001. <http://www.oasis-open.org/committees/relax-ng/>
- [32] J. Clark. *TREX - tree regular expressions for XML*. Thai Open Source Software Center Ltd, 2001. <http://www.thaiopensource.com/trex/spec.html>
- [33] A. Davidson, M. Fuchs, M. Hedin, M. Jain, J. Koistinen, C. Lloyd, M. Maloney, K. Schwartzhof. *Schema for Object-Oriented XML 2.0*. W3C Recommendation, 1999. <http://www.w3.org/TR/NOTE-SOX/>

- [34] A. Layman, E. Jung, E. Maler, H. S. Thompson, J. Paoli, J. Tigue, N. H. Mikula, S. De Rose. *XML-Data*. W3C Recommendation, 1998. <http://www.w3.org/TR/1998/NOTE-XML-data-0105/>
- [35] R. Jelliffe. *Resource Directory (RDDL) for Hook 0.2 : A One-Element Language for Validation of XML Documents based on Partial Order*. Academia Sinica Computing Centre, Taipei, 2002. <http://www.ascc.net/xml/hook/>
- [36] N. Klarlund, A. Møller, M. I. Schwartzbach. *Document Structure Description 1.0*. AT&T & BRICS, 1999. <http://www.brics.dk/DSD/specification.html>
- [37] D. Raggett. *Assertion grammars*. HPLabs, 1999. <http://www.w3.org/People/Raggett/dtdgen/Docs/>
- [38] C. Nentwich. *xlinkit Version 5 Language Reference*. Systemwire Ltd, 2002. <http://www.systemwire.com/xlinkit/Downloads/xlinkit-5.3/doc/reference/reference.html>
- [39] *Sun Multi-Schema XML Validator (MSV)*. Sun Microsystems, 1994-2003. <http://www.sun.com/software/xml/developers/multischema/>
- [40] K. Kawaguchi. *JARV User's Guide*. 2003. <http://iso-relax.sourceforge.net/JARV/JARV.html>
- [41] *XMLSpy*. Altova, 2003. <http://www.xmlspy.com>
- [42] *Java API for XML Processing (JAXP)*. Sun Microsystems, Inc., 1995-2003. <http://java.sun.com/xml/jaxp/>
- [43] *Simple API for XML (SAX)*. <http://www.saxproject.org/>
- [44] V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, A. Le Hors, G. Nicol, J. Robie, R. Sutor, C. Wilson, L. Wood, editors. *Document Object Model (DOM) Level 1 Specification, Version 1.0*. W3C Recommendation, 1998. <http://www.w3.org/TR/REC-DOM-Level-1/>
- [45] A. Le Hors, P. Le Hégarret, L. Wood, G. Nicol, J. Robie, M. Champion, S. Byrne, eds. *Document Object Model (DOM) Level 2 Core Specification, Version 1.0*. W3C Recommendation, 2000. <http://www.w3.org/TR/DOM-Level-2-Core/>
- [46] A. Le Hors, L. Cable, eds. *Document Object Model (DOM) Level 2 Views Specification, Version 1.0*. W3C Recommendation, 2000. <http://www.w3.org/TR/DOM-Level-2-Views/>
- [47] T. Pixley, editor. *Document Object Model (DOM) Level 2 Events Specification Version 1.0*. W3C Recommendation, 2000. <http://www.w3.org/TR/DOM-Level-2-Events/>
- [48] C. Wilson, P. Le Hégarret, V. Apparao, eds. *Document Object Model (DOM) Level 2 Style Specification, Version 1.0*. W3C Recommendation, 2000. <http://www.w3.org/TR/DOM-Level-2-Style/>
- [49] J. Kesselman, J. Robie, M. Champion, P. Sharpe, V. Apparao, L. Wood, eds. *Document Object Model (DOM) Level 2 Traversal and Range Specification, Version 1.0*. W3C Recommendation, 2000. <http://www.w3.org/TR/DOM-Level-2-Traversal-Range/>
- [50] J. Stenback, P. Le Hégarret, A. Le Hors, eds. *Document Object Model (DOM) Level 2 HTML Specification, Version 1.0*. W3C Recommendation, 2003. <http://www.w3.org/TR/DOM-Level-2-HTML/>
- [51] *Xerces*. The Apache Software Foundation, 1999, 2000. <http://xml.apache.org/xerces-j/>
- [52] R. L. Costello, et al.. *XML Schemas: Best Practices*. xFront, 2003. <http://www.xfront.com/BestPracticesHomepage.html>
- [53] J. Clark, S. DeRose, editors. *XML Path Language*. W3C Recommendation, 1999. <http://www.w3.org/TR/xpath>
- [54] S. Adler, A. Berglund, J. Caruso, S. Deach, T. Graham, P. Grosso, E. Gutentag, A. Milowski, S. Parnell, J. Richman, S. Zilles. *Extensible Stylesheet Language (XSL), Version 1.0*. W3C Recommendation, 2001. <http://www.w3c.org/TR/xsl>
- [55] *Xalan*. The Apache Software Foundation, 2003. <http://xml.apache.org/xalan-j/>
- [56] *MARC XML MARC Validation Stylesheets*. Library of Congress. <http://www.loc.gov/standards/marcxml///validation/genvalidate.xsl>
- [57] *MARC XML Official Web Site*. Library of Congress, 2003. <http://www.loc.gov/standards/marcxml//>
- [58] *marc4j*. CollabNet, Inc., 1999-2003. <http://marc4j.tigris.org/>
- [59] *Web Services Activity*. W3C Recommendation, 2002-2003. <http://www.w3.org/2002/ws/>