Interoperabilità tra elementi nei metadati bibliografici

Daniela Canali

Biblioteca comunale di Terni
daniela.canali@comune.terni.it

Nell'ambito dei più recenti studi sulla *mappatura* di metadati bibliografici, Carol Jean Godby, Devon Smith ed Eric Childress del centro ricerche del Online Computer Library Center (OCLC) propongono un modello computazionale per la formalizzazione del concetto di *mappatura*.[1] Tale processo separa la semantica dalla sintassi e specifica liste di mappature eseguibili dalla macchina focalizzate su asserzioni di equivalenza degli elementi e strettamente connessi con l'analisi intellettuale sottostante.
Gli autori hanno sviluppato un modello di dati, chiamato Morfrom, utilizzato come formato di metadati interno generico ed un linguaggio logico XML, chiamato Semantic Equivalence Expression Language (Seel): grazie a queste due componenti OCLC ha costruito un toolkit per la gestione di grandi collezioni di records di metadati, il Crosswalk Web Service.[2]
L'incontro tra l'utente ed il documento di interesse, nettamente facilitato dall'uso di cataloghi di biblioteche disponibili in Internet, risulta favorito quando questi forniscono indicazioni basate anche sull'abbinamento tra il luogo di residenza dell'utente e la facilità di reperimento del documento, andando ad indicare ad esempio le biblioteche più vicine alla sua abitazione che lo possiedono. Tale servizio, a notevole valore aggiunto, richiede un fluido scambio di dati e di mappature tra gli schemi di metadati utilizzati nelle varie realtà coinvolte: catalogo di ricerca, editori, singola biblioteca.
Si delinea uno scenario che presuppone una forte interazione tra la comunità bibliotecaria e gli editori. Il flusso di lavoro non è particolarmente complesso, né può dirsi limitato a biblioteche ed editori specializzati, dal momento che le transazioni avvengono su risorse che devono semplicemente essere trasferite da una comunità all'altra.
Il raggiungimento di tale obiettivo richiede che i due standard descrittivi, entrambi sofisticati, radicati nelle rispettive comunità di pratica, ed entrambi necessari in quanto annotano gli esiti di eventi che coinvolgono il libro in due contesti diversi, siano sincronizzati. Lo standard ONIX, ad esempio, utilizzato in ambito editoriale, ha tratto origine dalla necessità di identificare e tracciare i libri nel momento del loro acquisto e vendita. La transazione può coinvolgere una libreria online, tipo Amazon.com, che abbia il link ad un item fisico e alla relativa descrizione in linguaggio macchina, associando questo con materiali ausiliari, quali siti Web, guide di studi, indici, riviste o pagine digitalizzate, il tutto finalizzato a creare un contesto ricco per interagire con il libro prima e dopo l'acquisto.
Lo standard MARC si è invece evoluto dalla necessità di creare una descrizione autorevole, stabilendo responsabilità intellettuale, soggetto e provenienza di un lavoro interpretato come un record. Il record MARC in realtà ha poca attinenza con i dati delle transazioni commerciali o con gli elementi richiesti per popolare un certo sito Web, mentre un record ONIX è meno ricco di collegamenti alle liste di autorità per autore e soggetto. Ma le due descrizioni non sono incompatibili.

La mappatura

Tra ONIX e MARC esiste in realtà una certa sovrapposizione dal momento che nell'esempio le descrizioni si riferiscono ad uno stesso libro. Ma solo grazie ad un esperto di specifiche formali e dell'uso di entrambi gli standard è possibile sbrogliare le sovrapposizioni e registrarle in una tipica tabella di corrispondenze chiamata mappatura (*crosswalk*).[3]
La tabella 1 riporta un esempio di mappatura ONIX-MARC curata dalla Library of Congress:[4]

[1] Carol Jean Godby – Devon Smith – Eric Childress, *Toward element-level interoperability in bibliographic metadata*, "The Code4Lib Journal", 2 (2008), <http://journal.code4lib.org/articles/54>.
[2] <http://wiki.tdwg.org/twiki/pub/NCD/NCDWorkShop/OCLC_Crosswalk_Service.ppt>.
[3] Il termine *crosswalk* è utilizzato nel glossario dell'articolo *Introduction to Metadata: Pathways to Digital Information* (<http://www.getty.edu/research/conducting_research/standards/intrometadata/glossary.html>) in cui viene definito come segue: "A chart or table that represents the semantic mapping of fields or data elements in one data standard to fields or data elements in another standard that has a similar function or meaning. Crosswalks enable heterogeneous databases to be searched simultaneously with a single query as if they were a single database (semantic interoperability) and to effectively convert data from one metadata standard to another. See also metadata mapping below. Also known as field mapping.".
[4] *ONIX to MARC 21 Mapping*, <http://www.loc.gov/marc/onix2marc.html>.

Tabella 1 - Segmento di una mappatura da ONIX a MARC

Title composite <Title> ***The Title Composite contains the text of a title, including a subtitle when necessary.***		
<b202>	<TitleType>	(Used to set field 246 2nd indicator: If b202 = 00, 246 I2 = #; if b202 = 01, 246 I2 = 2)
<b276>	<AbbreviatedLength>	
<b203>	<TitleText>	246 $a
<b030>	<TitlePrefix>	245 $a
<b031>	<TitleWithoutPrefix>	245 $a
<b029>	<Subtitle>	245 $b; 246 $b

Se la finalità è condividere record tra biblioteche ed editori, è necessario stabilire tra ONIX e MARC una corrispondenza chiara tra gli elementi che identificano autori, titoli, editori, date di pubblicazione, identificatori numerici, edizioni, formati fisici, sommari, indici e soggetti. Meno chiaro appare cosa fare con le descrizioni di item raggruppati, come i siti Web associati all'item e le recensioni di utenti che possono avere a loro volta descrizioni bibliografiche complete. È comunque abbastanza semplice costruire corrispondenze nel lavoro quotidiano nelle biblioteche digitali e nelle organizzazioni a supporto delle biblioteche, compresa la creazione di record descrittivi, la normalizzazione ed il miglioramento, le ricerche su database e la generazione automatica di "bozze" di record d'archivio (inizializzazione) che vengono poi completate da esperti umani.
Non di meno gli autori hanno scoperto che persino le più piccole incompatibilità tra i formati dei record, possono creare condizioni di ostacolo alla soddisfazione delle richieste degli utenti. Considerando ad esempio che tra i pochi elementi dei principali standard utilizzati per rappresentare l'informazione bibliografica principalmente viene utilizzato il titolo del libro, nella mappatura MARC-Dublin Core, il campo 245 $a mappa al campo titolo Dublin Core-(DC) *title* – una relazione che persiste attraverso XML, ASN.1/BER,[5] RDF,[6] e vari altri schemi più o meno standard. Tale relazione si mantiene attraverso le diverse versioni di entrambi gli standard, così come OAI-PMH Dublin Core,[7] MARC-XML,[8] MARC 2709,[9] e le Versioni 1.1 e 1.2 del Unqualified Dublin Core (o DC-Simple), ed anche Dublin Core Terms[10] (o Qualified DC). A tale gruppo di schemi si può aggiungere anche lo standard MODS,[11] una semplificazione del MARC progettata per la descrizione delle risorse nella biblioteca digitale, in cui il concetto di *titolo* (*title*) – <titleinfo><title> – è mappato al 245 $a e al DC:title, secondo la mappatura della Library of Congress. Il MARC 245 $a è mappato anche su diversi elementi di ONIX, tra i quali <DistinctiveTitle>.
Inoltre sono state sviluppate mappature tra Dublin Core ed ONIX, nelle quali l'elemento DC:title corrisponde a <DistinctiveTitle>. Esistono quindi mappature tra MARC e Dublin Core, ONIX e MARC, ONIX e Dublin Core, MODS e MARC, MODS e Dublin Core.
Per un singolo concetto, ad esempio *title*, le relazioni tra gli standard sono coerenti e facili da implementare anche da parte di un profano. Ma nella forma attuale questa informazione è difficile da usare per l'elaborazione automatica, che richiede specifiche risorse tecniche, quali schemi XML e script XSLT,[12] troppo spesso non disponibili, non funzionanti o obsoleti.
La relativa scarsità di file in linguaggio di programmazione implica che è difficile trasformare questo contenuto in un formato eseguibile e mantenerlo. Ma forse alla base c'è anche la difficoltà dello step precedente, di analisi intellettuale. Non è infatti una funzione di poco conto determinare se le mappature esistenti sono sufficienti a realizzare una vera traduzione tra standard, o come possono eventualmente essere modificate o migliorate.
Spesso, quando è necessario sviluppare una mappatura, questa viene creata ex-novo anche se potrebbe essere derivata con algoritmi. È necessario un modello formale mi-

[5] *Introduction to ASN.1*, <http://asn1.elibel.tm.fr/introduction/index.htm>.
[6] Resource Description Framework (RDF), <http://www.w3.org/RDF>.
[7] Open Archives Initiative, <http://www.openarchives.org/>.
[8] MARCXML: MARC 21 Schema, <http://www.loc.gov/standards/marcxml/>.
[9] ISO: 2709:1996, <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=7675>.
[10] DCMI Metadata Terms, <http://dublincore.org/documents/dcmi-terms/>.
[11] MODS: Metadata Object Description Schema, <http://www.loc.gov/standards/mods/>.
[12] L'Extensible Stylesheet Language Transformations (XSLT) è un linguaggio basato su XML che si usa, insieme a un software in grado di interpretarlo, per trasformare i documenti XML. Sebbene ci si riferisca a questo processo come trasformazione, il documento originale non viene modificato; invece, viene creato un nuovo documento XML a partire da quello corrente. Poi il nuovo documento viene serializzato (output) dal processore nella sintassi dell'XML standard o in un altro formato come HTML o solo testo. Solitamente viene utilizzato per convertire i dati da un'applicazione XML a un'altra o per convertire un documento XML in una pagina web o un documento PDF. I file di questo formato sono essenzialmente file di testo, contengono elementi ed attributi ed hanno l'estensione ".xsl". L'XSLT è diventato uno standard Web con una direttiva (Reccomandation) W3C del 16 novembre 1999 (<http://www.w3.org/TR/xslt>).

gliore, che renda più facile l'analisi intellettuale, la sua associazione ai file eseguibili, la definizione di una proposta realistica per la gestione delle diverse versioni e la codifica degli standard.

Verso l'interoperabilità a livello di schema

Stando alla definizione di interoperabilità fornita nella prima parte del rapporto di Chan e Zeng[13] del 2006, il vero problema è ottenere una *schema-level interoperability* dal momento che si sta provando ad identificare il livello comune tra schemi di metadati formalmente definiti, indipendentemente dalla necessità a breve termine di processare record di richiesta. Scopo dello studio dei ricercatori OCLC è invece la creazione di un *framework* concettuale che renda semplice tale lavoro, creando una soluzione riusabile per i diversi tipi di funzioni e applicazioni.
Chan e Zeng identificano diversi oggetti concettuali allo scopo di ottenere questo tipo di interoperabilità: oltre alla mappatura, vengono descritte derivazioni, passaggi da uno schema all'altro e profili di applicazione. Una derivazione si ottiene quando un nuovo schema viene creato da uno precedente, come ad esempio MODS derivato da MARC al fine di creare uno standard più appropriato ai software di gestione delle descrizioni nelle collezioni digitali. Uno *switching-across schema* è un formato comune, definibile come nucleo interoperabile (*interoperable core*) in un modello cosiddetto *hub-and-spoke*[14] e in cui diversi schemi possono essere mappati per creare un modello di elaborazione estremamente efficiente. Con *profilo applicativo* si intende infine uno schema ibrido creato dalla combinazione di elementi da più di uno schema, osservando determinate restrizioni su come utilizzarli.[15] I profili applicativi permettono l'interoperabilità attraverso il riuso di elementi definiti in schemi esistenti e creando nuovi elementi solo quando necessario. Ad esempio, il Dublin Core Terms è un *profile* che estende il Dublin Core Element set con un ricco vocabolario per esprimere relazioni, date, livelli di pubblico. Mappature, derivazioni, modelli *hub and spoke* e profili applicativi rispondono alla necessità di identificare un livello comune nel complesso scenario della descrizione delle risorse. Tali oggetti implicano anche una tensione irrisolta tra l'esigenza di minimizzare la proliferazione di standard e quella di creare descrizioni di risorse utili ed elaborabili automaticamente. La ricerca OCLC si concentra sulle mappature, in linea con il contesto istituzionale e tecnico dell'organizzazione e con l'interesse di escogitare soluzioni riusabili. In realtà la mappatura può essere considerata alla base degli altri due oggetti analizzati da Chan e Zeng: le derivazioni richiedono infatti una corrispondenza per mappare gli elementi da uno schema madre al derivato e la stessa esigenza ha d'altra parte il modello *hub and spoke* per mappare gli schemi relativi ad un nucleo interoperabile.
La mappatura viene costruita a partire da un'unità di analisi fondamentale: un singolo elemento definito in uno standard, come ad esempio *title* in Dublin Core, MARC, MODS ed ONIX. Come le parole in un linguaggio naturale, i metadati possono essere estratti da un contesto e combinati in uno nuovo, formando profili di applicazione. Possono anche entrare in relazioni semantiche, come le equivalenze definite dalle mappature, che assomigliano alla sinonimia. L'interoperabilità a livello di schema fornita da Chan e Zeng riduce il problema dell'interoperabilità a livello di elemento, la cui infrastruttura tecnica è la base su cui possono essere eventualmente costruiti sistemi più complessi.
La ricerca OCLC descrive un modello computazionale generico che permette a chi ha una conoscenza approfondita di metadati usati nelle biblioteche, nei musei, in ambito educativo, nell'industria editoriale o in altri contesti che abbiano modelli di dati ben definiti, di sviluppare traduzioni creando una lista di equivalenze simile alla mappatura mostrata nella tabella 1.
Il modello è strutturato in modo che gli esperti di standard possano concentrarsi sulla mappatura di elementi equivalenti, lasciando che il software gestisca i dettagli disordinati che emergono dalle realizzazioni strutturali multiple, come quelle derivanti quando il concetto *title* viene identificato in MODS, MARC, Dublin Core ed ONIX.
Il modello somiglia moltissimo alle mappature prodotte dalla mente umana ed è quindi richiesta un'elaborazione minima per rendere tali dati attivi, dal momento che il processo di conversione delle mappature per l'esecuzione dei codici può essere ampiamente automatizzato: una volta che il sistema è impostato con le principali mappature, può funzionare con il minimo intervento umano, e sono richieste solo leggere modifiche per aggiungere nuovi standard o sistemare nuove versioni o traduzioni. Il sistema OCLC costituisce il nucleo logico del Crosswalk Web Service, risultato del programma di ricerca illustrato alla conferenza internazionale "Dublin Core (DC)-2003"[16] e successivamente nel-

[13] Lois Mai Chan – Marcia Lei Zeng, *Metadata Interoperability and Standardization - A Study of Methodology. Part I: Achieving Interoperability at the Schema Level*, "D-Lib Magazine", 12 (2006), 6, <http://www.dlib.org/dlib/june06/chan/06chan.html>.
[14] Espressione derivante dal linguaggio aeronautico usata per definire un modello di rete conosciuto anche come modello "a raggiera" o "a stella" nell'ambito delle compagnie aeree. Questo modello si diffuse dopo la liberalizzazione del trasporto aereo negli Stati Uniti. Le compagnie più grandi passarono quindi dal modello *point-to-point*, che prevede il collegamento di due città tramite un volo diretto, a quello *hub-and-spoke*, in cui tutti i voli che partono e arrivano dagli aeroporti secondari della rete (gli *spokes*) vengono convogliati verso un aeroporto principale, l'*hub*.
[15] Rachel Heery – Manjula Patel, *Application Profiles: Mixing and Matching Metadata Schemas*, "Ariadne", 25 (2000), <http://www.ariadne.ac.uk/issue25/app-profiles/>.
[16] Carol Jean Godby – Devon Smith – Eric Childress, *Two Paths to Interoperable Metadata*, "Dublin Core (DC)-2003: Supporting Communities of Discource and Practice–Metadata Research& Applications", Seattle, Washington (USA), September 28 - October 2 2003, <http://www.oclc.org/research/publications/archive/2003/godby-dc2003.pdf>.

l'articolo di Godby, Young e Childress.[17] Il sistema è attualmente usato per potenziare la funzionalità di mappare i metadati nell'OCLC Connexion® ed in molti servizi offline che elaborano grosse quantità di record. Una demo pubblica è accessibile dalla pagina OCLC's ResearchWorks.[18]
Il Crosswalk Web Service rappresenta un'importante componente del lavoro di reingegnerizzazione dei prodotti OCLC volto a creare una collezione di servizi modulari configurabili per la raccolta, la validazione, l'ampliamento, la manipolazione e la descrizione delle risorse, molti dei quali reingegnerizzati tramite software Open Source e resi disponibili tramite Web services. Ne consegue un'architettura accessibile via Internet che riduce la duplicazione interna degli sforzi e rivela la funzionalità critica a qualsiasi partner, cliente o parte che riconosca un set comune di standard.

Un modello di processo per la mappatura dei record

La figura 1 presenta gli step principali per separare la semantica dalla sintassi nel modello elaborato dal Crosswalk Web Service: le componenti possono essere riutilizzate e gli script di mappatura semplificati. Ma la conseguenza più significativa è che la funzione di creare ed interpretare le mappature rimane astratta, cosicché l'esperto di semantica dei metadati può concentrarsi sulla definizione delle mappature senza impantanarsi in complessi dettagli strutturali. Nell'esempio di figura 1 viene illustrato il processo di traslazione della descrizione del libro *Rocket Boys*[19] citato come esempio nello studio OCLC, codificata in formato ONIX nella figura 2. I record di input conformi allo standard di metadati di origine (stato A) vengono convertiti in una struttura XML comune (step 1) e quindi tradotti nella semantica dello standard di metadati target (step 2). Il risultato della mappatura è un set di record XML normalizzati (stato C) convertiti alla struttura nativa dello schema target (step 3), producendo come risultato record allo stato D. Solo il primo e l'ultimo passaggio del modello hanno accesso ai dettagli sintattici degli standard nativi. Nello step 2 intermedio le traduzioni operano come una struttura comune definita in locale.

Step 1: lettura dell'input

Un programma specifico, il *reader*, converte i metadati di un record di input nella struttura interna richiesta dal software di mappatura del modello. Questa struttura è un contenitore XML denominato Morfrom, la cui Document Type Definition contiene quattro elementi: il *record*, elemento a livello superiore; l'*header* che identifica un namespace di default per il record e linka ad una risorsa contenente una definizione degli elementi che lo popolano; il *field*, il cui attributo *name* identifica un elemento nello standard nativo, ed infine il *value* che definisce un percorso al dato. Gli elementi *field* conservano l'ordine dell'elemento, la sequenza e la struttura gerarchica, mentre gli attributi *name* conservano il nome degli elementi nel record nativo. Gli elementi *value* infine racchiudono i soli dati nel record.
La figura 2 mostra il risultato ottenuto dalla mappatura del record ONIX nel record Morfrom di OCLC: tale record contiene gli elementi autore, titolo e soggetto che dovranno essere mappati. I record Morfrom sono indicati per convenzione con il suffisso ".mfm".

Figura 1 - Il modello di mappatura

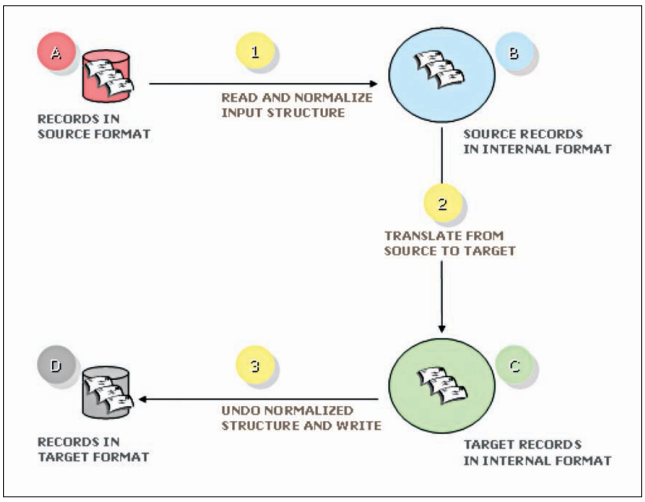
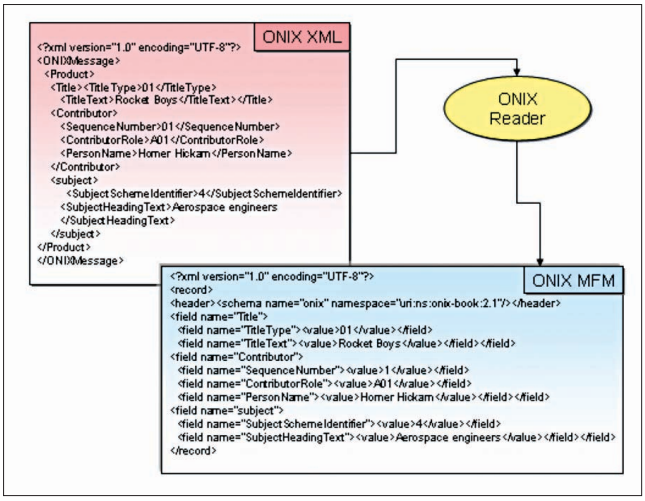


Figura 2 - Il record di input nella sintassi nativa ONIX e Morfrom



Step 2: la mappatura

La componente chiave del modello è rappresentata dallo step 2 della figura 1 dove avviene la mappatura. Tale pro-

[17] Carol Jean Godby – Jeffrey A. Young – Eric Childress, *A Repository of Metadata Crosswalks*, "D-Lib Magazine", 10 (2004), 12, <http://www.dlib.org/dlib/december04/godby/12godby.html>.
[18] <http://www.oclc.org/research/researchworks/default.htm>
[19] <http://www.homerhickam.com/books/rb.shtml>.

Figura 3 - Esempio semplice ma completo della scrittura Seel

```
view plaincopy to clipboardprint?
1.
2.  <?xml version="1.0" encoding="ISO-8859-1"?>
3.  <!-- dc:source = http://www.loc.gov/marc/onix2marc.html -->
4.  <translation>
5.  <header>
6.    <sourceschema name="onix" namespace="uri:ns:onix-book:2.1"/>
7.    <targetschema name="marc" namespace="uri:ns:marc:21"/>
8.  </header>
9.  <map id="1">
10.   <source>
11.     <mainpath>
12.       <branch><step name="Title"/><step name="TitleText"/></branch>
13.     </mainpath>
14.   </source>
15.   <target>
16.     <mainpath>
17.       <branch><step name="245"/><step name="a"/></branch>
18.     </mainpath>
19.   </target>
20. </map>
21. <map id="2">
22.   <source>
23.     <mainpath>
24.       <branch><step name="Contributor" position="1"/><step name="PersonName"/></branch>
25.     </mainpath>
26.   </source>
27.   <target>
28.     <mainpath>
29.       <branch><step name="100"/><step name="a"/></branch>
30.     </mainpath>
31.   </target>
32. </map>
33. </translation>
```

Figura 4 - Una mappatura Seel con un contesto

```
view plaincopy to clipboardprint?
1.
2.  <map id="3">
3.   <source>
4.    <mainpath>
5.      <branch bid='1'><step name="subject"/><step name='SubjectHeadingText'/></branch>
6.    </mainpath>
7.    <context bid='1'>
8.     <equals>
9.       <path from= '..'><step name='SubjectSchemeIdentifier'/></path>
10.      <value>4</value>
11.    </equals>
12.   </context>
13.  </source>
14.  <target>
15.   <mainpath>
16.     <branch bid='1'><step name='650'/><step name='a'/></branch>
17.   </mainpath>
18.   <context bid='1'>
19.     <equals>
20.       <path><step name='i2'/></path>
21.       <value>0</value>
22.     </equals>
23.   </context>
24.  </target>
25. </map>
```

cesso ha due input: la parte logica ed un record codificato in Morfrom nello standard dei metadati di origine, ad esempio il record che contiene gli elementi ONIX mostrato nella parte inferiore della figura 2. La parte logica della mappatura è scritta nel linguaggio XML Seel (Semantic Equivalence Expression Language) definito da OCLC, elaborato da un traduttore personalizzato che esegue le funzioni fondamentali di XSLT o altri semplici linguaggi di scrittura. Rappresenta un modello computazionale della mappatura che mantiene la caratteristica di definizione, le asserzioni di equivalenza, nell'applicazione attiva al primo livello. La figura 3 presenta una scrittura Seel completa inglobata nell'elemento <translation> che ha due *figli*: l'<header>, che specifica gli standard di metadati nell'origine e nel target della mappatura, e uno o più elementi <map>, ognuno dei quali contenente approssimativamente le stesse informazioni, ossia un elemento definito nello standard di metadati di origine ed il corrispondente elemento nel target. La mappa Seel è modulare, autocontenuta ed astratta dalla sintassi nativa dello standard. Può inoltre essere in formato leggero o completo, secondo il numero delle mappe che contiene. La figura 3 mappa ONIX <Title> <TitleText> con MARC 245 $a e la prima occorrenza di ONIX <Contributor> <PersonName> con MARC 100 $a.

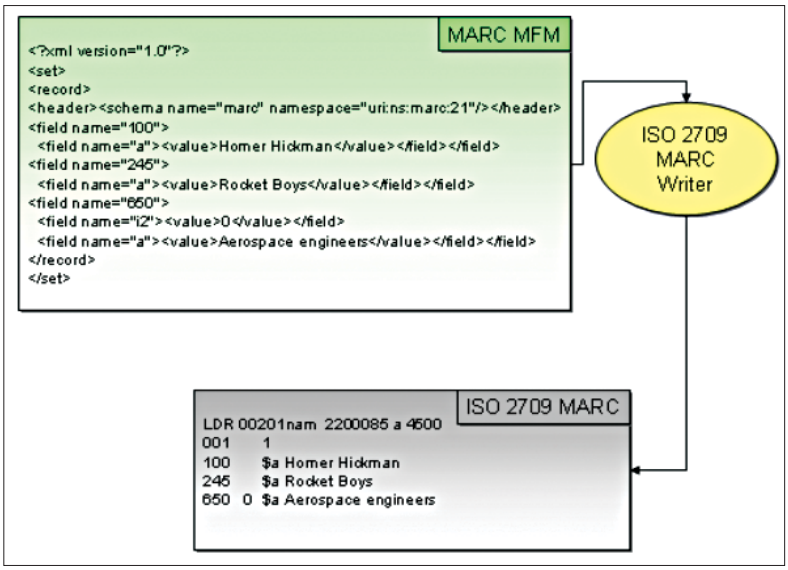
In termini più tecnici, Seel mappa gli elementi che possono corrispondere ad un concetto di *title* nei due standard usando l'informazione nell'elemento Seel <mainpath><branch>, che identifica un percorso nel record di origine e costruisce un percorso equivalente nel target. Una volta che i percorsi sono definiti, i dati coinvolti nella mappatura sono trasferiti dal linguaggio di markup della fonte a quello del target. Per far ciò Seel ha un linguaggio di percorso: ogni elemento Seel <step name='abc'> individua un elemento Morfrom <field name='abc'> e definisce uno step su un percorso dal record di origine ai dati. Per esempio, la prima mappa Seel della figura 3 localizza il frammento <field name='title'><field name='title Text'> in un record Morfrom in ONIX e costruisce il percorso <field name='245_><field name='a'> nel corrispondente record Morfrom in MARC.

La figura 4 mostra la mappa Seel per l'elemento *subject*. Si tratta di una mappa più complessa rispetto a quella della figura 3, in quanto l'obiettivo è mappare l'elemento <subject> di ONIX <SubjectHeadingText> con un'intestazione di soggetto della Library of Congress in MARC 650 $a if; la fonte ONIX ha un elemento <SubjectSchemeIdentifier> *fratello* il cui valore è 4, codice ONIX per la Library of Congress Subject Heading in questo contesto. Questa mappa costruisce anche un campo MARC i2 e lo popola. Tutto questo lavoro viene fatto dall'elemento <context> che ha due essenziali sotto-elementi: un <path> che descrive un percorso relativo a quello descritto nel corrispondente elemento <mainpath>; ed un <value> che contiene i dati.

Nella <source>, l'elemento <context> implementa la parte 'if' della dichiarazione data sopra, localizzando il sotto-albero che contiene il '4'. L'attributo 'form' nel percorso usa il linguaggio della sottodirectory UNIX per indirizzare qualsiasi percorso nel record, e qui specifica che <Subject SchemeIdentifier> è un derivato (*fratello*) di <subject> <SubjectHeadingText> che può essere verificato controllando il record Morfrom mostrato nella parte in basso della figura 2. Nel <target> può comparire anche un <context> interpretato come un'istruzione per la costruzione di un percorso aggiuntivo, un *figlio* della radice definita nel <mainpath>. Nella mappa mostrata nella figura 4, l'elemento <context> è usato per creare un indicatore ed asse-

gnargli i dati appropriati; ne risulta l'elemento <field name='650'> mostrato nel record Morfrom nella parte alta della figura 5.
La mappa mostrata nella figura 4 è un frammento di una mappa più estesa che costruirebbe gli elementi MARC *subject* con diverse tag e strutture interne dipendenti dal fatto che il codice SubjectSchemeIdentifier sia LCSH, MESH, o uno schema unico di un editore. Nella scrittura completa ci sarebbero molteplici elementi <branch> e <context> limitati ad un altro con l'attributo 'bid' (*branch id*).

Figura 5 - Un record Morfrom scritto in ISO MARC



Readers e *writers*

L'esempio sopra descritto ha condotto alla mappatura di un record di input in ONIX XML in uno di output MARC ISO per dimostrare che il modello OCLC Crosswalk Web Service è sufficientemente flessibile da poter gestire record in XML e non, purché formalmente definiti e coerenti. I *readers* ed i *writers* invocati agli step 1 e 3 richiedono un'elaborazione personalizzata ma ripagano lo sforzo con un livello maggiore di astrazione.
Tali programmi, generalmente scritti in Java, nascondono dettagli complessi e talvolta strani relativamente alla struttura gerarchica, all'ordine degli elementi e alla sintassi interna rispetto al resto del modello. E poiché tali dettagli sono proprio quelli che tendono a mutare quando lo standard evolve, *readers* e *writers* possono essere scritti una sola volta e poi riutilizzati ampiamente. Il Crosswalk Web Service ne possiede ora una collezione in grado di manipolare o generare Morfrom MARC per ISO 2709 MARC, MARC-XML, MARC-8[20] e formati basati su MARC sviluppati in locale. Lo step di normalizzazione sintattica semplifica anche la mappatura. Come mostrano le figure 2 e 5, la rappresentazione Morfrom elimina molte differenze superficiali tra ONIX e MARC e quindi la complessità delle regole richiesta per mappare i due standard. Ed è necessario un solo set di mappature, poiché *readers* e *writers* gestiscono la variazione strutturale.
La medesima mappa potrebbe quindi funzionare per MARC XML o per MARC SO-2079 verso ONIX o Dublin Core Terms. Questa riusabilità si ottiene a spese di procedure di elaborazione extra rispetto a quando la sintassi nativa del record è XML, ma, visti i vantaggi, il costo è minore.

Figura 6 - Parte di una stylesheet XSLT per la relazione ONIX-MARC

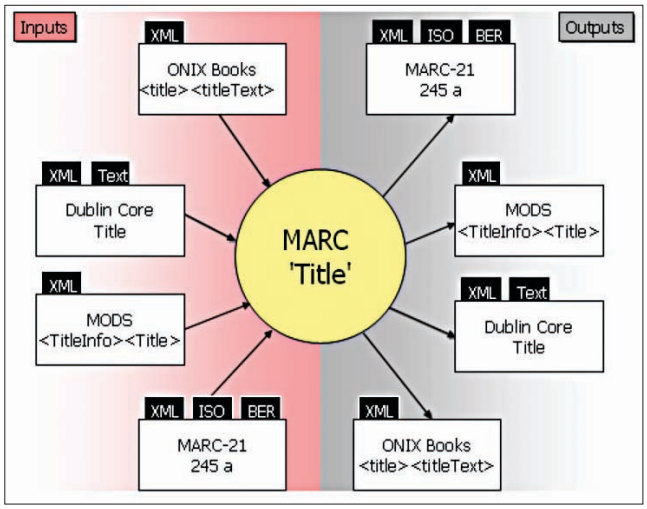
```
view plaincopy to clipboardprint?
1.
2.      <xsl:for-each select="contributor[PersonName]">
3.        <xsl:call-template name="datafield">
4.          <xsl:with-param name="tag">100</xsl:with-param>
5.          <xsl:with-param name="ind1">1</xsl:with-param>
6.          <xsl:with-param name="ind2">#</xsl:with-param>
7.          <xsl:with-param name="subfields">
8.            <subfield code="a">
9.              <xsl:value-of select="."/>
10.           </subfield>
11.         </xsl:with-param>
12.       </xsl:call-template>
13.     </xsl:for-each>
```

La mappatura: perché non farla in XSLT?

A questo punto si potrebbe obiettare sull'opportunità del motore di conversione Seel rispetto ad alternative più popolari, quali la stylesheet XSLT che traduce direttamente dall'origine al target.[21] La figura 6 mostra un frammento di una stylesheet XSLT che costruisce MARC 100 1# *$a* da ONIX <contributor><PersonName> e fa il medesimo lavoro dello script Seel mostrato nella figura 5.[22] Questo esempio deriva dalla stylesheet ONIX-MARC mantenuta dalla Library of Congress,[23] molto semplificata in questa sede.
Questa stylesheet è un set di istruzioni per costruire un record MARC XML che contiene campi e sottocampi propriamente formattati nell'ordine richiesto da MARC. Le istruzioni per costruire e popolare il campo 100 mostrato qui sono situate nello script completo tra quelli che creano i campi di controlli e quelli per i campi variabili

[20] MARC 21 Specifications for Record Structure, Character Sets, and Exchange Media, <http://www.loc.gov/marc/specifications/specchartables.html>.
[21] Cfr. nota 12.
[22] In realtà, gli script non sono analoghi poiché questo script inserisce indicatori, mentre la mappa Seel corrispondente nella figura 5 non fa altrettanto. Ma la mappa Seel mostrata in figura 6 mostra come si possa fare.
[23] ONIX to MARCXML Stylesheet, <http://www.loc.gov/standards/marcxml/xslt/ONIX2MARC21slim.xsl>.

Figura 7 - Un modello di mappatura hub-and-spoke *per il cencetto di* title'



in un record MARC valido. La funzione di costruzione dei campi è effettuata contestualmente a quella di mappatura del record: paragonato al modello OCLC, uno script Seel come quello mostrato nella figura 6 è molto meno astratto poiché combina il lavoro svolto da uno script Seel con quello fatto da *readers* e *writers*. Per produrre output non XML sarebbe naturalmente richiesto un processo aggiuntivo in un modello che dipende da XSLT per tradurre i record di metadati. Diversamente dal corrispondente script Seel, la stylesheet XSLT è di conseguenza asimmetrica, poiché l'origine ed il target hanno codifiche qualitativamente diverse. Il percorso di origine è definito usando gli elementi dell'XPath XSLT,[24] che identifica la tag ONIX di interesse nella Linea 1 e trasferisce i dati esistenti in quel nodo al record target nella Linea 8. Ma il percorso del target è definito più semplicemente da un set di elementi XML rigidamente codificati ottenuti dalla specifica del record target. Per esempio, la Linea 3 stabilisce la radice del percorso target trasferendo un valore ad una *template* XSL che produce l'elemento <datafield tag='100_>. La Linea 7 estende questo percorso con l'elemento <subfield code= 'a'>, una relazione che non è esplicitamente etichettata, come potrebbe essere in una dichiarazione Seel, ma è desunta dal fatto che questa linea è eseguita immediatamente dopo la Linea 3. Una mappa Seel non è mai ambigua perché tutto è perfettamente etichettato con gli elementi definiti nella Document Type Definition di Seel.

Dal momento che una mappa Seel è esplicitamente etichettata e simmetrica, può essere riversata tramite algoritmo trasponendo i contenuti degli elementi <source> e <target>. Nelle mappature "uno a uno" mostrate nella figura 3 questa relazione è forse ovvia, ma è anche vera nella mappa più complessa della figura 4. Nella mappatura opposta, un MARC 650 $a contenente un secondo indicatore con il valore '0' scatena la costruzione di ONIX <subject><SubjectHeadingText> e costruisce un elemento *fratello* <SubjectSchemeIdentifier> con il valore '4'. La caratteristica di reversibilità di Seel è ancora in fase di studio presso OCLC ma tali risultati si sono già rivelati utili nello sviluppo e nella valutazione di traduzioni biunivoche.

Messa a punto dello step di mappatura

Il modello di mappatura definito dal Crosswalk Web Service di OCLC in questo contesto ha caratterizzato la mappatura da ONIX a ISO 2709 MARC. Questo è uno dei possibili percorsi del sistema ma ne esistono molti altri. Per incentivare la riusabilità il Crosswalk Web Service implementa un modello *hub-and-spoke*[25] che può richiedere due step di mappatura: uno per mappare il record di input ad un formato intermedio (*core format*) ed un secondo per mappare questo con l'output desiderato. Tale formato, che non ha alcuna esistenza reale a livello di implementazione di software o struttura di dati, è nient'altro che una convenzione per la scrittura di un primo set di script Seel con un target comune (lo script *to-*) ed un secondo set con una fonte comune (lo script *from-*). In un servizio progettato per incontrare le esigenze di una comunità bibliotecaria, un *core format* basato su MARC produce notevoli risparmi poiché le mappature sono divise in input *to-MARC* ed output *from-MARC*. Quando viene introdotto un nuovo input *to-MARC* come un MODS, questo ha accesso automatico alla collezione esistente di tutti gli output *from-MARC*. Come risultato, una volta che la collezione di input ed output cresce anche se moderatamente, il modello *hub-and-spoke* elimina un considerevole sforzo di elaborazione. La versione base del Crosswalk Web Service dispone dei seguenti output: quattro sintassi di codifica MARC ed estensioni di MARC definite localmente, cinque versioni e due sintassi di codifica di Dublin Core, due versioni di MODS ed una versione rispettivamente di ONIX-Books ed ONIX-Serials. Quando viene aggiunto l'input MODS-to-MARC, tutti questi output diventano disponibili, comprese le mappature MODS-ONIX e MODS-Dublin Core Terms, per cui gli esperti di metadati hanno tentato di scrivere mappature, forse non necessarie, a causa delle molte mappature rilevanti implicate. Questo modello può facilmente risolvere i problemi relativi alla mappatura di *title* sopra discussa. La figura 7 mostra un ipotetico modello di mappatura *hub-and-spoke* che mappa un singolo elemento e mostra le sintassi di codifica, rappresentate dalle etichette nere, che potrebbero essere gestite dai *readers* e dai *writers*. Complessivamente dovrebbero essere richieste solo tre mappe *to-MARC* e due mappe *from-MARC*. Ma in una richiesta di mappatura di un set di metadati ad un target standard e alla relativa codifica il modello può gestire 49 diverse combinazioni di input ed output. Naturalmente, il

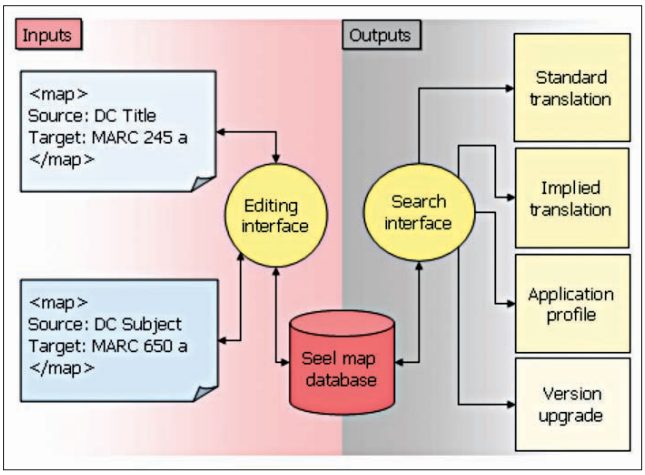
[24] XML XPath language (XPath), <http://www.w3.org/TR/xpath>.
[25] Cfr. nota 14.

Crosswalk Web Service completo ha una collezione di script complessi che mappano centinaia di elementi. Ma la figura 7 potrebbe accuratamente rappresentare le relazioni di mappatura nel sistema completo se vengono omessi tutti i riferimenti agli elementi '*title*'.
Il problema di scrivere mappature per ogni input ed output non sussiste dal momento che la collezione di traduzioni cresce. Il modello OCLC separa la semantica del problema – l'asserzione di equivalenza tra coppie di elementi – dalla sintassi, eliminando una varietà di realizzazioni che dovrebbero essere decodificate all'inizio del processo e riassemblate alla fine.

Le mappature Seel in una risorsa persistente

Un set di mappe produce una mappatura in una forma calcolabile attraverso gli *statements* del linguaggio dichiarativo. Le relazioni di interesse sono asserzioni di equivalenza semantica, che rappresentano la valutazione da parte di esperti di standard di metadati secondo cui coppie di *author*, *title* ed altri elementi in due standard sono sinonimi, o almeno così simili che la differenza esistente non inficia il risultato della mappatura. Poiché una mappa Seel è completamente etichettata, può essere interpretata sia come set di istruzioni eseguibili sia come dati statici, potendo così funzionare come un record in un database di relazioni tra standard di metadati omogenei. Questi record di mappatura possono inoltre essere raccolti, ricercati, assemblati dinamicamente e persino estratti per nuove informazioni. La figura 8 mostra una rappresentazione schematica di un database di mappe Seel accessibile tramite un servizio associato al Crosswalk Web Service.
Un esperto di standard di metadati inserisce una nuova linea in un form elettronico di una mappatura, tipo un foglio di calcolo Excel. Poiché una mappa Seel assomiglia moltissimo ad un'intestazione in una mappatura, nella maggior parte dei casi può essere generata automaticamente. L'esperto di standard e lo sviluppatore di software hanno solo bisogno di osservare poche convenzioni comuni per codificare la fonte, il target e speciali condizioni quali i valori dell'indicatore, che appare negli elementi <context> di Seel.
Una volta che queste convenzioni sono state stabilite, un piccolo set di cambiamenti può costituire il solo sforzo richiesto per aggiornare una mappatura in una nuova versione dello standard, dal momento che molte delle mappe nel database possono essere riutilizzate. Allorché le nuove mappe sono state generate ed aggiunte al database, l'esperto interessato alla traduzione ONIX-MARC può usare l'interfaccia di ricerca per recuperare tutte le mappe con una fonte ONIX ed un target MARC. Si tratta di una tipica mappatura standard-standard, ma nel modello OCLC è concettualmente simile ad altri output.

Figura 8 - Mappe Seel in una risorsa persistente



Una query di ricerca potrebbe ad esempio specificare fonti Dublin Core o ONIX e target MARC, producendo una mappatura che imita un *application profile*. Un'altra ricerca potrebbe produrre tutte le mappe Seel il cui target sia MARC 245 $a. Se l'origine in una mappa nel set di risultati identifica il percorso ONIX <DistinctiveTitle>, mentre l'origine in un'altra identifica il percorso MODS <titleinfo> <title>, un processo automatico potrebbe produrre una mappatura sicura tra i campi *title* in ONIX e MODS, usando MARC 245 $a come intermediario. Il database delle mappe Seel tiene conto di alcune importanti realtà riguardo alla mappatura di metadati come oggetti concettuali per ottenere l'interoperabilità a livello di elementi di cui parlavano Chan and Zeng.[26]
Le mappe corrispondenti alla mappatura pubblicata tra una coppia di standard possono essere recuperate dal database con una semplice query di ricerca, come spiegato prima. Con molte probabilità, invece, la mappatura avrà bisogno di essere personalizzata in qualche modo, aggiungendo, cancellando o eliminando mappe particolari per favorire le esigenze di una specifica funzione locale. La logica di scomposizione della traduzione in mappe e del loro immagazzinamento emerge dalla considerazione che la maggior parte delle mappe deve essere persistente. Quindi, una volta mappata, la relazione tra Dublin Core Title e MARC 245 $a, essa viene memorizzata e riutilizzata in una varietà di traduzioni e codifiche strutturali o per estrarre informazioni.

Uso del servizio di traduzione

La traduzione logica è disponibile per le applicazioni OCLC tramite il Crosswalk Web Service, e per applicazioni di terzi attraverso un link sulla pagina Web del gruppo di ricerca.[27] Il servizio appare all'applicazione client come una "scatola nera" che elabora un ordine di lavoro espres-

[26] Cfr. nota 13.
[27] Un prototipo è disponibile all'indirizzo <http://www.oclc.org/research/researchworks/default.htm>.

so nel Web Services Description Language,[28] in cui l'utente fornisce una tripletta che specifica lo standard di metadati, la codifica della sintassi e del carattere per l'input e l'output della traduzione. Il servizio viene invocato da molti contesti, di diversa complessità, da processi di *batch*, che traducono solo pochi elementi, ad un'interfaccia di editing che permette ad un catalogatore di inserire un record dettagliato e di vederlo in formati multipli, in tempo reale. Le mappature sono ottenute da fonti pubbliche come la Library of Congress e da esperti di metadati di OCLC.
Le mappature create internamente sono inviate come fogli di calcolo. Nel corso dello sviluppo del servizio, le mappature codificate come fogli di calcolo hanno rappresentato una difficoltà anche per gli esperti di soggettazione in OCLC. Per ottenere, ad esempio, la relazione tra Dublin Core e MARC, l'esperto ha mappato Dublin Core Terms e MARC, da cui è stato generato automaticamente uno script Seel. Poiché Dublin Core Terms è un profilo applicativo, il principio di semplificazione che collassa le estensioni come DCTerms:Title:Alternative in DC:Title può essere utilizzato per generare la mappatura a Unqualified o Simple Dublin Core. Lo stesso input intellettuale produce due traduzioni diverse, molte di più quando si tiene conto di diverse codifiche strutturali, ed ancora di più quando queste traduzioni vengono usate come origine o destinazione nella relazione che coinvolge MARC e MODS, GEM o ONIX. Ciò è possibile perché nel modello di traduzione OCLC ogni componente è progettata per il riuso.
In futuro OCLC ha intenzione di snellire la scrittura di mappature attraverso lo sviluppo di un'interfaccia utente che accetti gli input richiesti dalla struttura della mappa Seel e generi mappature o script Seel umanamente leggibili. Il lavoro di OCLC è focalizzato sui record bibliografici ma si discute sull'opportunità di set aggiuntivi di traduzioni per rappresentare la descrizione di liste di autorità, proprietà e dati museali. L'obiettivo è creare un ambiente in cui sia il più semplice possibile sfruttare le potenzialità delle mappature per qualunque tipo di markup. Attraverso l'uso ripetuto, le mappature saranno controllate minuziosamente e riviste e quelle più popolari finiranno per emergere come standard *de facto*.

Stato dell'interoperabilità a livello di schema

Lo studio descrive un modello di traduzione, un modello di dati (Morfrom), una specifica di linguaggio (Seel), un *toolkit* software (Crosswalk Web Service) ed una *demo* pubblica per la gestione su larga scala di diversi flussi di metadati, partendo dall'ipotesi che emergono vantaggi reali se l'unità atomica di analisi è una mappa, o una singola equivalenza tra due elementi e non una traduzione completa tra due standard, soluzione tipicamente proposta nell'ambito dei problemi di mappature di metadati.
In breve, il Crosswalk Web Service rende più semplice implementare l'interoperabilità a livello di elemento e permette di normalizzare i record gestiti quotidianamente da OCLC. Tali record rappresentano circa una dozzina di standard di metadati definiti ed una vastissima gamma di diversità strutturali. Ma poiché OCLC serve una specifica comunità di pratica, la semantica delle relazioni richieste per eseguire un notevole numero di transazioni è generalmente ben chiara e ben compresa. Ad un livello più basso dell'elemento sussiste un insieme di problemi – normalmente definiti di normalizzazione di campo – come ad esempio la standardizzazione di dati contenuti in un elemento particolarmente critici per la gestione dei record bibliografici, quali nomi, date e citazioni. In questi casi, come nel tipico record bibliografico che passa attraverso i sistemi OCLC, l'espressione sintattica può essere diversa ma la semantica è trasparente. Come risultato, OCLC sta esplorando la potenzialità di un linguaggio di scrittura simile a Seel che operi a livello di sotto-elemento e che possa essere scritto come un set persistente di istruzioni recuperate da un database per aggiungere, cancellare o cambiare la disposizione dei dati.
Ad un livello superiore rispetto a quello dell'elemento, esiste invece il problema più complesso di aumentare l'interoperabilità di due standard rendendo più coerenti i loro modelli astratti. I problemi affrontati nel Crosswalk Web Service risiedono per lo più nell'ambito dell'ingegneria del software, mentre riconciliare le differenze tra i modelli astratti costituisce un problema sociologico molto più complesso che richiede analisi intellettuale, consulenza con i "custodi" degli standard ed un compromesso bilaterale, come avvenuto ad esempio con il modello concettuale CIDOC[29] usato per descrivere oggetti museali tramite vocabolari controllati sviluppati per applicazioni ben più ampie nelle comunità bibliotecarie digitali.[30] La risoluzione definitiva del problema richiederebbe un'ontologia di quanto ritenuto importante per la descrizione di una vasta gamma di risorse e generi. E l'ontologia delle descrizioni continua in realtà ad evolversi, dopo il lavoro d'avanguardia con cui sono state definite le componenti essenziali nelle descrizioni dei diritti di proprietà intellettuale o di materiali educativi sviluppati in contesti tecnici ed istituzionali complessi. Dal momento che OCLC non si è ancora cimentata con i metadati per oggetti museali, i problemi descritti da Doerr[31] potrebbero apparire remoti. Invece problemi simili risiedono già nel nucleo di ONIX e di MARC: malgrado molte coppie di elementi dei due standard possano essere mappate e rese accessibili attraverso il Crosswalk Web

[28] Web Services Description Language (WSDL), <http://www.w3.org/TR/wsdl20/>.
[29] The CIDOC Conceptual Reference Model, <http://cidoc.ics.forth.gr/>.
[30] Carl Lagoze – Jane Hunter, *The ABC Ontology and Model*, "Journal of Digital Information", 2 (2001), 2, <http://jodi.tamu.edu/Articles/v02/i02/Lagoze/lagoze-final.pdf>.
[31] Cfr. nota 27.

Service, sono molto più numerose quelle che aspettano ancora la risoluzione di un problema il cui punto di snodo è che la mappatura autorevole mantenuta dalla Library of Congress è troppo ambigua per essere usata come input diretto nel sistema OCLC. Inoltre lo standard ONIX per i libri definisce un record che descrive un oggetto in primo piano ed un set opzionale di oggetti accessori – recensioni, indici, siti Web, materiale promozionale e persino pezzi d'abbigliamento, ognuno dei quali può essere accompagnato da una descrizione dettagliata che include un titolo – mentre un record MARC è progettato come descrizione di un singolo item.
È quindi quanto mai necessario il lavoro collaborativo tra le due istituzioni di alto livello per risolvere le differenze tra i modelli astratti che sottostanno ai due standard.

Abstract

Within the latest studies on bibliographic metadata crosswalking, Carol Jean Godby, Devon Smith ed Eric Childress of the Research Center of Online Computer Library Center (OCLC) propose a computational model for crosswalking concept formalization. The authors describe a translation model, a data model (Morfrom), a language specification (Seel), a software toolkit (Crosswalk Web Service) and a public demo for management of different metadata flows, assuming that tangible benefits arise if the atomic unit is a map or a unique equivalence between two elements and not a full translation between two standards, as typically proposed about metadata crosswalking issues.