

## **6. APPLICAZIONE SPERIMENTALE DI CDSWARE RELATIVA AL MODULO BIBFORMAT**

Il permesso di fare copie digitali o fisiche di tutto o parte di questo lavoro per uso di ricerca o didattico è acconsentito senza corrispettivo in danaro, mentre per altri usi o per inviare a server, ridistribuire a liste di discussione o diffondere ulteriormente è necessario il permesso da parte dell'autore.

L'utilizzo per scopi di profitto non è consentito senza il permesso dell'autore.

Gli eventuali lavori derivanti dallo stesso dovranno contenere opportuna citazione.

### **6.1 INTRODUZIONE**

Il presente capitolo rappresenta la parte sperimentale vera e propria dell'intero elaborato di tesi derivante dalla collaborazione avvenuta con i membri dello SBA (Servizio Bibliotecario d'Ateneo) di Messina.

Col loro appoggio si è realizzata un'applicazione del software CDSware relativa al problema della migrazione dei record bibliografici immagazzinati nel sistema di automazione di biblioteca Aleph 500 al sistema documentario CDSware, successivamente da loro adottato.

E' importante osservare come il compito assegnato dal dottor Feminò dello SBA abbia consentito la realizzazione di un sistema di gestione dei record che prende spunto da quello già realizzato presso il CERN di Ginevra dai membri dello staff di CDSware, descritto al paragrafo 5.3.

All'inizio del lavoro si è utilizzata l'ultima release disponibile on line, cioè la versione 0.1.2, che in seguito è stata abbandonata, poiché instabile e mancante di alcuni dei moduli presenti nell'unica (al momento in cui si scrive il presente elaborato) versione stabile 0.0.9, successivamente adottata per lo sviluppo della tesi.

La soluzione a tale compito, dopo una prima analisi di tipo generale dell'applicazione, ha coinvolto l'utilizzo di due moduli interni al software di

CDSware: BibConvert e BibFormat.

BibConvert assume il compito di effettuare la conversione dei record importati da Aleph in CDSware, rispettivamente dal formato interno UNIMARC al formato MARC 21. Tale parte dell'esperienza dello SBA è stata affrontata e trattata nella tesi di Buttà Basilio [B3].

BibFormat ha invece il compito sostanziale di creare un formato generale di rappresentazione, a livello di interfaccia web, dell'output dei record trasferiti da Aleph a CDSware.

La descrizione e l'analisi di tale modulo, che trae spunto dalle informazioni presenti nella documentazione interna all'installazione del prodotto, rappresenta la prima parte del presente capitolo, dove vengono presentate le diverse funzionalità e possibilità di personalizzazione attraverso interfacce web, ma soprattutto attraverso la programmazione in linguaggio EL (Evaluation Language), progettato dagli implementatori del sistema.

Infine, la seconda parte presenta la soluzione, relativamente al modulo BibFormat, creata per risolvere il compito richiesto dallo SBA.

## **6.2 BIBFORMAT**

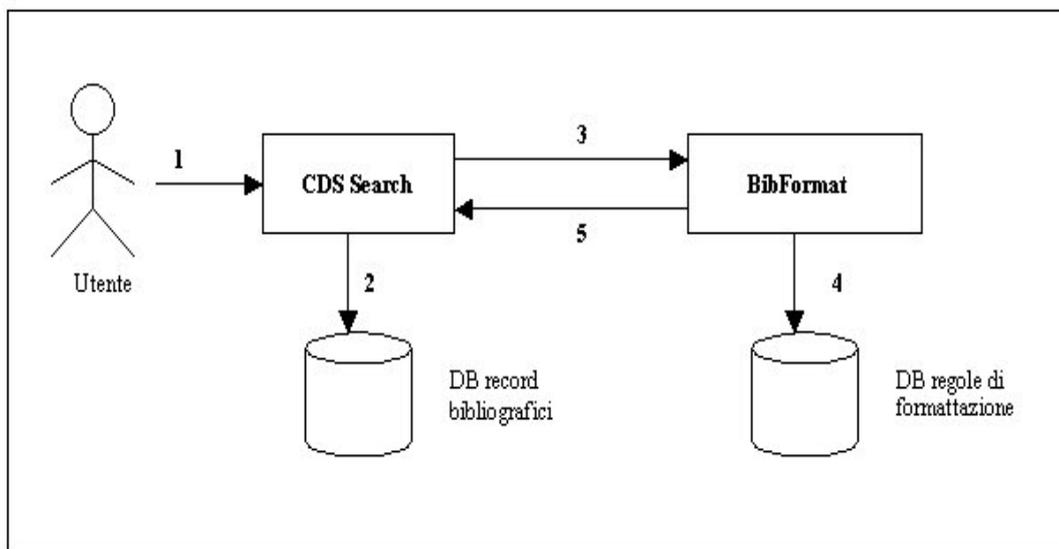
*BibFormat* è l'applicazione interna al modulo *CDS Search Weblib* che si occupa della ricerca dei metadati presenti nel repository dell'applicazione CDSware.

Il suo compito è quello di fornire un meccanismo flessibile di formattazione dei record bibliografici, mostrati come risultato delle query rivolte da parte dell'utente, permettendo agli amministratori o agli utenti stessi di personalizzare l'aspetto della loro presentazione.

Il modulo offre anche la possibilità di usare un *sistema di linking* in grado di generare automaticamente tutti i link presenti nei record (di accesso al fulltext di una risorsa, di riferimento ad un periodico elettronico, ecc.) riducendo in maniera

considerevole la gestione manuale dei link stessi da parte di chiunque voglia renderli disponibili nella rappresentazione di output.

La seguente figura 1 mostra, a grandi linee, il funzionamento del modulo in questione:



**Figura 1 – Fasi di elaborazione del modulo BibFormat**

Come si può notare, il modulo BibFormat opera secondo i seguenti passi:

1. una query dell'utente é ricevuta dal modulo di ricerca CDS search weplib

2. weblib ottiene i record che corrispondono alla query, dal database che li contiene
3. weblib chiede a BibFormat (modulo interno a weblib stesso, ma separato nello schema per motivi esplicativi) di formattare i record ottenuti in risposta alla query
4. bibFormat controlla il suo database delle regole di formattazione e, per ogni record, determina quale formato deve essere associato, applica il formato e risolve i link possibili
5. bibFormat restituisce tutti i record formattati a weblib, il quale realizza una pagina HTML che include i risultati formattati

E' importante notare come in tale processo di elaborazione BibFormat permetta la separazione della “presentazione” dei risultati di ricerca dalla “logica” interna usata dal motore di ricerca vero e proprio per reperire i record richiesti. Ciò è reso possibile dalla disponibilità di un ulteriore repository contenente le regole di formattazione memorizzate per ogni singolo record.

Per determinare le regole di formattazione, denominate *behaviors* (comportamenti), di BibFormat è possibile utilizzare un'interfaccia web di configurazione ed il *linguaggio EL* (Evaluation Language) opportunamente progettato dal gruppo di CDS, che permettono di creare i propri comportamenti personalizzati e successivamente di applicarli o, per meglio dire, associarli ai vari record.

Per quanto visto, BibFormat prende in input i record bibliografici presenti nel sistema e ne produce in output una versione *formattata e lincata*.

Per versione formattata si intende la possibilità che il modulo produca una versione trasformata dei record di ingresso: ad esempio una rappresentazione in

HTML degli stessi, ovvero, una trasformazione interamente specificabile a priori dall'utente.

Per lincata, invece, si intende la possibilità di indicare a BibFormat di includere riferimenti a risorse Internet collegate al dato bibliografico nella versione formattata del record. Tale inclusione deve seguire alcune regole preconfigurate. Attualmente il modulo tratta in input record XML MARC 21 presenti in un file di testo, ma le versioni future permetteranno di trattare direttamente altre modalità di input: acquisizione da database, chiamate a funzione, ecc..

### **6.2.1 Interfaccia web di configurazione**

L'interfaccia di configurazione di BibFormat consiste in un insieme di pagine web che presentano all'utente i principali aspetti di configurazione dell'applicazione; sono a disposizione, infatti, differenti sezioni navigabili ognuna delle quali corrispondente ad una proprietà di configurazione di BibFormat.

Le sezioni disponibili sono le seguenti:

- *OAI Extraction Rules*: qui sono definiti i tipi e le regole di mappatura per i record OAI MARC XML di input. E' infatti in tale sezione che si trovano le informazioni sulle *variabili interne* e le loro corrispondenze con i tag XML dei record bibliografici. I nomi delle variabili possono essere successivamente usati nelle regole di formattazione e di linking. Ad esempio si può indicare che il campo "100 \$a" deve essere mappato nella variabile interna "\$100.a", che potrà essere usata in seguito
- *Behaviors*: è la sezione principale contenente le definizioni di differenti tipi di output o comportamenti preconfigurati, che permette di indicare a BibFormat come deve comportarsi quando un tipo di output da applicare

ad uno o più record è selezionato dall'utente. I comportamenti vengono passati poi come parametri del modulo quando esso esegue la formattazione dei metadati

Ad esempio, in questa fase, si può dire a BibFormat di creare un file contenente i record di formato da associare ai record bibliografici separatamente, o di creare un file arricchito in cui siano presenti contemporaneamente le informazioni bibliografiche e quelle di formattazione dei record

- *Link Rules*: permettono di accedere al repository contenente le regole di definizione dei link che dicono a BibFormat come generarli per i riferimenti all'interno dei record.

Ad esempio si può definire una regola per creare un link ad un database "Persone" dalla variabile interna "\$100.a", definita e opportunamente mappata in precedenza attraverso le regole di estrazione, che rappresenta il nome dell'autore

- *UDFs*: tale sezione presenta una lista di tutte le funzioni definibili dall'utente che possono essere adoperate dentro le espressioni dell'Evaluation Language (EL) usate per specificare differenti aspetti di configurazione. Permette anche di modificare o estendere tale lista creando nuove funzioni, eventualmente derivanti dal riutilizzo di quelle predefinite, consentendo perciò l'esecuzione di formattazioni complesse senza alterare il codice di programma di BibFormat.

Ad esempio si può definire una funzione che permetta la ricerca e l'estrazione di indirizzi email da un file di testo

- *Formats*: è un'utile caratteristica di BibFormat per definire una porzione di codice EL a cui è possibile assegnare un'etichetta per consentirne il riutilizzo. Con molta probabilità questa funzionalità è quella maggiormente configurata e permette il riutilizzo di formati, UDF, ecc..

Ad esempio si può specificare che l'autore venga stampato in corsivo, che in caso di più di 10 autori solo i primi 3 vengano stampati, ecc.

- *KBs*: è l'interfaccia di definizione dei cosiddetti Knowledge Bases, utili per garantire la rappresentazione univoca di un dato comune a più record bibliografici, ma indicato al loro interno con stringhe di denominazione diverse. I KBs successivamente alla definizione, diventano disponibili all'interno delle espressioni di linguaggio EL.

Ad esempio permette di specificare che Phys Rev D e Physical Review D rappresentano entrambi lo stesso periodico e che dovrebbero essere standardizzati come Phys Rev : D

- *formati dei file* (File Formats) – definisce i tipi di formati dei file basandosi sulle estensioni. Ad esempio si può dire all'applicazione che i file \*.pdf, dove per \* si indica qualsiasi nome con estensione .pdf, saranno trattati come file in formato PDF (Portable Document Format)

I prossimi paragrafi descriveranno più in dettaglio le sezioni presentate, in quanto strettamente necessarie per la comprensione del funzionamento del modulo BibFormat.

### **6.2.2 OAI Extraction Rules (Regole di estrazione)**

Questa sezione dell'interfaccia web di BibFormat permette di definire le cosiddette *variabili interne*, utilizzate successivamente dal motore dell'applicazione e dalle altre sezioni per contenere i valori dei tag MARC 21 presenti nei record di metadati di input.

Quella che viene svolta in questa fase é una sorta di mappatura che permette di definire la corrispondenza dei tag di input in nomi di variabile immagazzinabili

ed utilizzabili da BibFormat per le fasi successive.

Come già detto, BibFormat prende alcuni record bibliografici di input e, in accordo ad un comportamento pre-configurato, li formatta, ad esempio in una rappresentazione HTML.

I record di metadati possono arrivare in ingresso in diversi formati: record di database, convenzioni XML differenti dall'XML MARC 21 ecc..

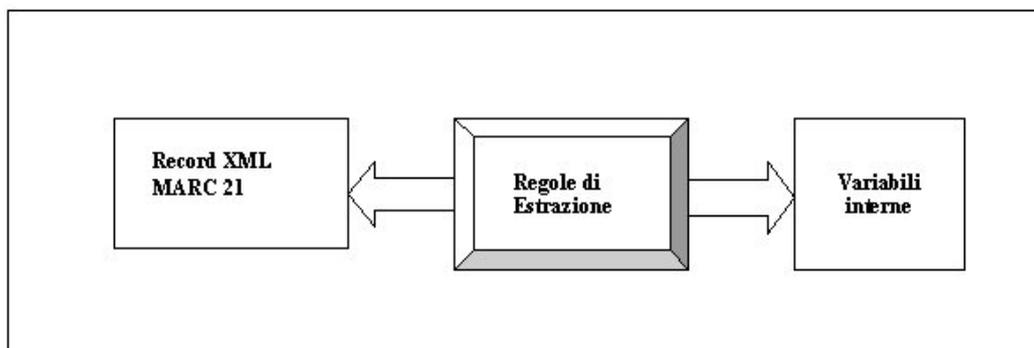
Le variabili interne, anche se al momento solo per input di formato XML MARC 21, esistono proprio per questo motivo, in quanto forniscono una maniera comune per riferirsi ai dati di input senza tener conto di alcun concreto formato per la loro rappresentazione.

La possibilità data dalle regole di estrazione è quella di poter successivamente definire i link ed i comportamenti di BibFormat riferendosi alle variabili interne che, dal punto di vista dell'applicazione, rappresentano i tag di un record bibliografico nel formato MARC.

Da ciò segue che, in linea generale, è possibile usare i comportamenti di BibFormat per qualunque formato di input che possa essere mappato nelle variabili interne dell'applicazione.

La figura 2 seguente mostra a grandi linee il meccanismo di estrazione espresso

prima:



**Figura 1 – Meccanismo di estrazione delle variabili interne dai record XML MARC 21**

Una variabile interna rappresenta quindi un’etichetta da assegnare ad un particolare tag di input. Essa può avere dei campi, anch’essi rappresentabili come etichette, che corrispondono ai campi dei tag MARC21 di input: ad esempio una variabile interna può mappare il tag MARC 21 “100 \$a” nelle etichette di variabile “\$100.a” e “\$100.e” indicando i campi corrispondenti a \$a e \$e che rappresentano rispettivamente il nome e l’editore della risorsa.

Le variabili e i loro campi supportano i valori multipli.

Da quanto appena detto si capisce come basandosi su un input in formato XML MARC i concetti di variabile e campo possano essere facilmente ricondotti a quelli di tag e di campo della struttura di un record MARC.

Dato il seguente record:

```
<oai_marc>

  <varfield id="037" i1="" i2="">

    <subfield label="a">SCAN-0009119</subfield>

  </varfield>

  <varfield id="100" i1="" i2="">

    <subfield label="a">Amelotti, Ercole</subfield>

  </varfield>

  <varfield id="100" i1="" i2="">

    <subfield label="a">Buttà, Basilio</subfield>

    <subfield label="e">editor</subfield>

  </varfield>

</oai_marc>
```

si può osservare come:

- Ogni occorrenza dell'elemento *varfield*, avente il compito di indicare il valore del tag MARC 21 del record, corrisponderà sempre ad una differente variabile interna estratta
- Ogni occorrenza del sottocampo interno ad un dato elemento *varfield* di un record, corrisponderà ad un differente valore di campo della variabile che mappa l'elemento *varfield*.

Bibformat avrà così un insieme di regole che dicono a quale nome di variabile corrisponde un elemento *varfield* e a quale nome di campo di variabile

corrisponde l'elemento sottocampo di un tag MARC 21.

Attraverso l'interfaccia web é possibile aggiungere, cancellare e modificare variabili interne e relativi campi.

La seguente tabella mostra un esempio di come BibFormat esegua la *definizione di mappatura* tra alcune variabili e i campi di input:

<b>Etichetta variabile</b>	<b>Tag di mappatura</b>	<b>Valore multiplo</b>	<b>Sottocampi</b>
\$100.a	<varfiled id= "100" i1 = "" i2 = "">	Yes	<subfield label= "a">
\$100.e	<varfiled id= "100" i1 = "" i2 = "">	Yes	<subfield label= "a">
\$909C0.b	<varfiled id= "909" i1 = "C" i2 = "0">	Yes	<subfield label= "b">

Come si può vedere, la variabile "\$100" corrisponde al tag "100" e specificandola nei modi "\$100.a" e "\$100.e" BibFormat associa i corrispondenti sottocampi "a" ed "e". Specificando invece il tag "909" con la variabile "\$909C0.b" viene creata la corrispondenza con il relativo sottocampo "b".

Adesso si può vedere come BibFormat risolva i tag dei record bibliografici in nomi di variabile assegnando i valori contenuti nei primi.

Dato il seguente record di input assumiamo che la variabile associata al tag "100" ammetta valori multipli, mentre quella associata al tag "909" no:

<oai\_marc>

<varfield id="037" i1="" i2="">

<subfield label="a">SCAN-0009119</subfield>

</varfield>

<varfield id="100" i1="" i2="">

<subfield label="a">Racah, Giulio</subfield>

</varfield>

<varfield id="100" i1="" i2="">

<subfield label="a">Guignard, G</subfield>

<subfield label="e">editor</subfield>

</varfield>

<varfield id="909" i1="C" i2="0">

<subfield label="b">11</subfield>

</varfield>

<varfield id="909" i1="C" i2="0">

<subfield label="b">12</subfield>

</varfield>

</oai\_marc>

la risoluzione dei valori conseguenti alla mappatura dei tag “100” e “909” è mostrata nelle tabelle seguenti:

<b>Variabile “\$100”</b>		
valore 1	valore campo “a”	Racah, Giulio
valore 2	valore campo “a” valore campo “e”	Guignard, G editor

<b>Variabile "\$909C0"</b>		
valore 1	valore campo "b"	11

Bisogna notare come l'elemento varfield 037 del record non viene risolto perché non esiste un'entry corrispondente nella configurazione di BibFormat.

Inoltre se la variabile consente valori multipli come il tag “100”, si avranno due valori assegnati alla variabile “\$100”, rispettivamente “valore 1” e “valore 2” della prima tabella, mentre nel caso della variabile associata al tag “909C0” nella seconda tabella, benché il record contenga due tag 909, solo l'ultimo valore del tag sarà preso in considerazione da Bibformat.

### **6.2.3 Behaviors (comportamenti)**

I *comportamenti* rappresentano il passo finale della definizione del formato di rappresentazione dei record bibliografici. Infatti, prima di un comportamento, devono essere stabiliti i formats, le regole di link e i knowledge bases che saranno descritti nei paragrafi successivi.

Essi dunque rappresentano una sorta di etichetta di formato, specificabile attraverso la corrispondente sezione di interfaccia web, che sarà alla fine possibile applicare ad uno o più record di metadati.

Quando si richiede a BibFormat di formattare un insieme di record bibliografici, bisogna specificare il tipo di *rappresentazione di output* o *behavior*, consistente in una stringa che etichetta uno specifico formato.

La definizione di un comportamento determina la possibilità di poter specificare uno di due *tipi* a disposizione:

1. *Normal*: consiste in un comportamento che produce un record di output contenente solo le informazioni di formattazione da applicare ai record bibliografici.
2. *Input Enrich* (solo per input XML): crea un formato arricchito del record di input mettendo le impostazioni di formattazione alla fine del record stesso. In tal caso ogni record inserito nel database, avrà il proprio formato di rappresentazione incorporato al suo interno

Di seguito sono presentati due record di formattazione prodotti da BibFormat nelle due modalità Normal e Input Enrich:

### Record tipo Normal

<record>

```
<controlfield tag="001">000001189</controlfield>
```

```
<datafield tag="FMT" ind1="" ind2="">
```

```
<subfield code="f">hb</subfield>
```

```
<subfield code="g">&lt;hr&gt;&lt;table width=&quot;90%&quot;&gt;
```

```
&lt;tr&gt;&lt;td valign=&quot;top&quot;
```

```
align=&quot;left&quot;&gt;&lt;b&gt;&lt;/b&gt;&lt;br
```

```
&gt; c1998&lt;/td&gt; &lt;td valign=&quot;top&quot;
```

```
align=&quot;right&quot;&gt; &lt;br&gt; &lt;font
```

```

                size=-2&gt;&lt;b&gt;Â©CERN
                Geneva&lt;/b&gt;&lt;/font&gt;&lt;BR&gt;
                &lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;
    </subfield>
</datafield>
</record>

```

## Record tipo Input Enrich

```

<record>
  <datafield tag="001" ind1="" ind2="">000001189</datafield>
  <datafield tag="005" ind1="" ind2="">20020320094625.0</datafield>
  <datafield tag="020" ind1="" ind2="3">
    <subfield code="a">88-15-06306-4</subfield>
  </datafield>
  <datafield tag="041" ind1="0" ind2="">
    <subfield code="a">ita</subfield></datafield>
  <datafield tag="044" ind1="" ind2="">
    <subfield code="a">IT</subfield></datafield>
  <datafield tag="245" ind1="1" ind2="">
    <subfield code="a">Le trappole del welfare</subfield>
    <subfield code="c">Maurizio Ferrera</subfield>
  </datafield>
  <datafield tag="260" ind1="" ind2="">
    <subfield code="a">Bologna</subfield>
    <subfield code="b">Il Mulino</subfield>
    <subfield code="c">c1998</subfield>
  </datafield>
  <datafield tag="300" ind1="" ind2="">
    <subfield code="a">168 p.</subfield>
    <subfield code="c">21 cm</subfield>
  </datafield>
  <datafield tag="490" ind1="1" ind2="">
    <subfield code="a">Contemporanea</subfield>
    <subfield code="c">99</subfield>
  </datafield>
  <datafield tag="500" ind1="" ind2="">
    <subfield code="a">Segue: Appendice</subfield>

```

```

</datafield>
<datafield tag="760" ind1="0" ind2="">
  <subfield code="g">99</subfield>
</datafield>
<datafield tag="246" ind1="1" ind2="4">
  <subfield code="a">Uno stato sociale sostenibile per l'Europa del
  21. secolo</subfield>
</datafield>
<datafield tag="653" ind1="0" ind2="">
  <subfield code="a">Europa</subfield>
</datafield>
<datafield tag="653" ind1="0" ind2="">
  <subfield code="a">Welfare state</subfield>
</datafield>
<datafield tag="100" ind1="1" ind2="">
  <subfield code="a">Ferrera, Maurizio</subfield>
</datafield>
<datafield tag="040" ind1="" ind2="">
  <subfield code="a">IT SBA Messina</subfield>
  <subfield code="e">RICA</subfield>
</datafield>
<datafield tag="980" ind1="" ind2="">
  <subfield code="a">PICTURE</subfield>
</datafield>
<datafield tag="FMT" ind1="" ind2="">
  <subfield code="f">hb</subfield>
  <subfield code="g">&lt;strong&gt;&lt;/strong&gt; </subfield>
</datafield>
<datafield tag="FMT" ind1="" ind2="">
  <subfield code="f">hd</subfield>
  <subfield code="g">&lt;table border=&quot;0&quot;
    width=&quot;100%&quot;&gt;&lt;tr
    class=&quot;blocknote&quot;&gt;
&lt;td valign=&quot;left&quot;&gt;Pictures
&lt;small&gt; /
&lt;/small&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;
&lt;br&gt;&lt;p&gt;&lt;center&gt;&lt;br&gt;c1998

&lt;br&gt;&lt;small&gt;Il Mulino: Bologna

```

```

</small></center><p
style="margin-left: 15%; width:
70%">><br><small><strong>
;Note: </strong>Segue: Appendice;
</small><br><p> </subfield>
</datafield>
</record>

```

Si può notare come il record di formattazione Normal riferisca il record bibliografico "000001189" d'archivio attraverso il tag "001" e indichi a BibFormat la rappresentazione da applicare attraverso il tag "FMT" personalizzato di CDSware. Da ciò ne proviene la necessità di caricare separatamente i due record (bibliografico e di formattazione) nel caso si vogliano importare nel sistema.

Diversamente, il record di formattazione Input Enrich, oltre ai suddetti tag "001" e "FMT", contiene anche i tag informativi presenti nel record bibliografico. Questo record, perciò, costituisce una versione di quello bibliografico, arricchita semplicemente dal tag "FMT" indicante il formato di rappresentazione. In tal caso, per l'importazione del record nel sistema, basterà caricare il record di formato arricchito.

Un behavior si compone di una o più *condizioni* che dicono a BibFormat come realizzare il behavior ad essi relativo.

Ogni condizione può contenere a sua volta 0 o più *azioni* associate.

Una condizione é descritta da un'espressione di *Evaluation Language* paragrafo 6.2.8 che può dare un risultato booleano con valore "True" o "False".

Un'azione é un'espressione scritta in *Evaluation Language* che, quando la condizione associata assume valore "True", produce una serie di operazioni determinanti la formattazione di output del record bibliografico trattato e, quando invece assume valore "False" ne produce altre.

Quando BibFormat deve formattare un insieme di record con una data etichetta di comportamento esegue i seguenti passi:

1. controlla le condizioni di comportamento valutandone il codice EL “ordinatamente”: partendo da quella con numero d’ordine assegnato più basso fino ad arrivare a quella con numero più alto
2. controlla le azioni associate da svolgere quando una delle condizioni risulta avere “True” come risultato
3. valuta le azioni nell'ordine specificato, come per quanto detto per l’ordinamento delle condizioni e, infine, concatena i loro risultati.

Usando differenti condizioni si possono specificare formati alternativi dentro un comportamento: ad esempio è possibile formattare un record in maniera differente in dipendenza del valore dei suoi tag.

A scopo d’esempio possiamo considerare le seguenti condizioni:

**Condizione1: \$980.a=”PICTURE” → Azione1**

**Condizione2: “=” → Azione2**

Esse determinano l'esecuzione dell’Azione1 se il record di input possiede un tag “980 \$a” (tag personalizzabile nella notazione MARC21) con valore “PICTURE”, oppure l’Azione2, valida per qualsiasi altro tag, se “\$980” non é presente nel record.

Di seguito è mostrato un semplice esempio di come definire un comportamento per le proprie necessità:

si vogliono formattare dei record bibliografici, ma, dipendentemente dal numero associato al tag personalizzabile “909C0 \$b” (informazioni circa la pubblicazione per CDSware), si applicano azioni di formattazione differenti:

- se il record possiede un tag “909” contenete il valore “27”, verrà visualizzato il titolo in grassetto, tag “245”, e tutti gli autori secondari
- altrimenti, se il tag “980” non è presente, verrà visualizzato il titolo in “strong” e verranno stampati tutti gli autori principali

Per far ciò bisognerà definire da interfaccia un nuovo comportamento di tipo Normal che può essere chiamato ad esempio SIMPLE. Si dovranno definire così due condizioni esprimibili in linguaggio EL: una per applicare il formato a seguito della presenza del tag 909 e l'altra per il formato di default: nel caso in cui tale tag non sia presente nel record.

Il comportamento risultante dovrebbe essere definito utilizzando il seguente codice EL:

**Condizione1: \$909C0.b = “27”**

**Azione1:**

```
“<b>” $245.a ”</b>”  
forall($0248.a)  
{  
  “ – “ $700.a ”<br>”  
}
```

**Condizione2:** “”=””

**Azione2:**

```
“<strong>” $245.a ”</strong>”  
forall($100.a)  
{  
  “ – “ $100.a “<br>”  
}
```

Come si può vedere, la Condizione1 sarà valutata per prima da BibFormat: è importante infatti l'ordine in cui le condizioni, ma anche le azioni, vengono scritte. Nell'esempio è anche presente il costrutto “forall” che esegue in maniera ciclica la stampa di ciò che è racchiuso tra parentesi graffe: nella prima azione stamperà gli autori secondari, nella seconda quelli principali.

Tale costrutto ed il linguaggio EL sono descritti al paragrafo 6.2.9.8.

#### **6.2.4 Formats (fomati)**

Questa sezione dell'interfaccia web di BibFormat è di utilità per la programmazione in codice EL, in quanto permette di raggruppare attraverso un *identificatore unico* porzioni di codice EL che possono essere richiamate e quindi riutilizzate all'interno di altre espressioni del linguaggio, operazione peraltro descritta al 6.2.9.6.

In sostanza, il funzionamento è del tutto identico a quello delle chiamate di procedura dei linguaggi di programmazione procedurali.

In tale sezione é possibile visualizzare, modificare e cancellare format già definiti, oppure crearne di nuovi.

E' bene precisare come i format siano differenti dalle funzioni *UDFs* definite dall'utente descritte al paragrafo 6.2.6, in quanto quest'ultime utilizzano il PHP come linguaggio per il proprio codice.

L'utilizzo di tale sezione è alquanto semplice poiché basterà scrivere il codice in una “text area” apposita ed assegnare un'etichetta allo stesso tramite una “input box”.

### **6.2.5 Knowledge bases (KBs)**

Questa sezione dell'interfaccia web di configurazione di BibFormat, presenta la possibilità di mappare un insieme di stringhe preimmagazzinate, dette knowledge bases, in un'unica stringa anch'essa memorizzata precedentemente. Tale funzionalità permette quindi di far sì che determinati valori di input presenti all'interno dei record, vengano *normalizzati*, in maniera tale da ricadere sotto la stessa rappresentazione all'interno del repository di CDSware.

Un esempio per mostrare l'utilità di quanto appena detto é il seguente:

e' possibile indicare a BibFormat che “Phys Rev D” e “Physical Review D” sono entrambi lo stesso periodico, di modo che inserendo una qualunque delle due stringhe di rappresentazione, il nome del periodico sarà rappresentato in output nella forma standardizzata “Phys Rev : D”.

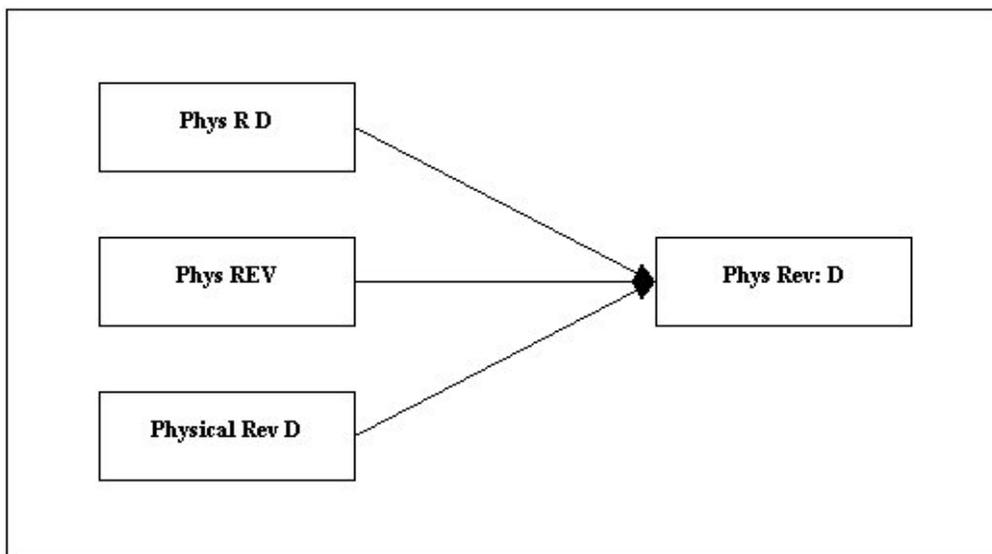
In tal modo chiunque esegua una ricerca o inserisca dei metadati relativi al periodico in questione, avrà la certezza che i record di metadati ad esso relativi potranno riferirsi in maniera univoca al periodico stesso.

La definizione di un Knowledge Base implica l'inserimento di una label univoca

per esso nell'interfaccia web e di tutte le possibili stringhe di input da mappare in un'unica stringa di output, anch'essa da specificare. Gli identificatori non sono case-sensitive.

Una volta definito, un Knowledge Base potrà essere opportunamente utilizzato all'interno di espressioni di codice EL come può essere visto al paragrafo 6.2.9.5.

La seguente figura 3 mostra in maniera semplificata come la mappatura viene realizzata:



**Figura 2 – Mappatura delle parole chiave attraverso i Knowledge Bases**

## 6.2.6 User Defined Functions (UDFs)

Le *UDF* (User Defined Functions) rappresentano una delle caratteristiche più potenti di BibFormat che consente l'utilizzo di operazioni o meglio funzioni, definite dall'utente, applicabili sulle stringhe.

Per affrontare questo compito BibFormat definisce un meccanismo che permette oltre che la definizione delle UDF, anche la possibilità di usarle dentro qualsiasi espressione del linguaggio EL.

Tali funzioni sono identificate da un nome univoco e possono ricevere dati di input nella forma di parametri.

Esse devono essere scritte con il linguaggio di programmazione PHP, che permette proprio di effettuare numerose operazioni di manipolazione sulle stringhe di testo.

E' possibile gestire in modo completo le UDF attraverso la sezione di interfaccia web "UDFs" in cui é presente una lista completa di tutte le UDF definite con il loro identificatore, i parametri, e una breve documentazione di cosa fanno.

Si possono anche aggiungere, cancellare, e modificare le funzioni o controllare il codice PHP di una funzione già definita.

La creazione di una funzione implica le seguenti considerazioni:

- Quando si vuole aggiungere o modificare una UDF viene richiesta una lista di parametri: si devono così inserire i nomi dei parametri separati da virgole.

Ad esempio se si vuole definire una nuova funzione per prefissare una data stringa con un'altra, è necessario inserire due parametri: la stringa che deve essere prefissata es: "Open" e quella che rappresenta il prefisso stesso es: "Archive"

Entrambi i parametri dovranno essere inseriti nella input box corrispondente.

- L'ordine con il quale si specificano i parametri di una funzione è l'ordine con il quale vengono passati all'UDF da un'espressione EL, dunque, bisogna far attenzione a non invertirli poiché si otterrebbero risultati sbagliati
- Quando si definisce il codice php di una funzione, ci sono altre cose importanti da considerare:
  - Il risultato di una funzione deve essere una stringa
  - I parametri sono disponibili dentro il codice php come variabili con il nome di parametro: ad esempio se il parametro si chiama "pippo", sarà possibile accedere al suo valore attraverso la variabile php "\$pippo"
  - Bisogna assicurarsi che il codice php sia corretto: non c'è modo di sapere se il codice è corretto attraverso BibFormat, poiché l'applicazione non prevede alcuna possibilità di analisi

Il paragrafo 6.2.9.4 più avanti mostrerà come è possibile invocare UDFs da espressioni di codice EL.

## **6.2.7 Links**

BibFormat, oltre che formattatore di record, è anche un *gestore di link*.

Per gestione dei link si intende la possibilità di definire un insieme di regole che descrivano come generare un link a partire da informazioni presenti nel record bibliografico.

Se il link può essere generato dalle suddette regole, allora il gestore potrà effettuare i seguenti controlli:

- controllare che il link sia valido
- controllare, per i link a file, che il file stesso esista, in quali formati; ecc. ed infine ritornare il link risolto

Dunque, se si ha a disposizione un insieme di record bibliografici che contengono certi link codificabili in regole note al gestore, non sarà necessario memorizzare ogni link per ogni record, in quanto sarà il gestore stesso a generarli dinamicamente.

Per far ciò è necessario gestire un piccolo set di regole e non migliaia di link statici per i vari record.

BibFormat permette di configurare, tramite la relativa sezione appartenente alla sua interfaccia web, differenti definizioni di link ognuna identificata da un nome univoco. Ogni definizione di link avrà poi alcuni parametri associati rappresentanti le informazioni passate alle regole stesse.

Quando in seguito si invoca il gestore per risolvere un link, ad esempio da espressioni di EL all'interno di un behavior o di un format, si dovrà specificare l'identificatore della definizione di link e il valore di ogni parametro usato dalla stessa definizione. Tali valori sono sempre stringhe.

Il gestore recupererà le regole associate alla definizione specificata e interpreterà quelle regole usando i valori di parametro passati, avvisando infine se il link è stato correttamente generato e, se sì, restituendo la sua risoluzione come risultato.

Attraverso l'interfaccia web é possibile accedere al repository delle regole per controllare quali sono le definizioni di link disponibili, o per definire nuove regole o gestire quelle già definite. Quando si aggiunge o modifica una definizione di link si dovranno specificare i parametri che dovranno essere separati da virgole.

Le regole dei link sono strutturalmente simili ai behavior. Benché ve ne possano essere differenti tipi, una definizione é composta da una o più condizioni e ognuna di esse può avere una o più azioni che dicono come il link deve essere costruito in caso la condizione sia verificata.

Le definizioni dei link, che includono condizioni e azioni, hanno una particolare struttura e sono descritte in linguaggio EL con una restrizione: l'espressione "EL LINK" non può essere usata per l'assegnazione dei nomi agli stessi.

Ogni gruppo di condizioni e azioni associato ad un link può essere di un differente "tipo risolvete": quando si crea una nuova definizione di link sarà richiesto tale tipo.

La struttura di formazione e il modo in cui il gestore interpreterà un link all'interno di un record dipende dal tipo risolvete che può essere "External" o "Internal".

Entrambi i tipi sono discussi nei paragrafi successivi.

Dato che una definizione di link é formata da più condizioni, quando una risoluzione é richiesta, il gestore di link recupera tutte le condizioni associate ed effettua le seguenti operazioni:

- prende la prima di esse seguendo il numero d'ordine di valutazione: la condizione col numero di linea associato più basso, sarà valutata per prima
- valuta l'EL della condizione e verifica il suo stato:

- se essa risulta essere “True”, le azioni associate vengono eseguite ed infine il link viene generato e restituito in una variabile a disposizione all’interno del codice EL, dando la possibilità ad esempio di renderlo disponibile in output e di seguire il link attraverso un click del mouse
- nel caso in cui risulti essere “False”, invece, controlla la successiva con valore di linea più alto
- se tutte le condizioni a disposizione risultano non verificate allora il gestore avverte che il link non può essere risolto

Questo comportamento é di tipo generale, ma il modo per determinare se un link é stato risolto o meno e la costruzione del link stesso dipende dal tipo risolvete.

### **6.2.7.1 Tipo di link External**

Questo tipo risolvete é il più semplice e viene usato quando si vuole che BibFormat generi un link che punta ad una risorsa esterna, ad esempio una pagina web.

I tal caso la condizione del link é composta da una sola azione che sarà valutata se la condizione associata risulta “True”. L'azione corrispondente sarà così eseguita ed il risultato di essa sarà restituito come link risolto in una variabile a disposizione all’interno del codice EL quando il gestore avrà finito l'elaborazione.

### 6.2.7.2 Tipo di link Internal

Questo tipo risolvete *Internal* è usato quando si vuole effettuare un link ad un documento: sostanzialmente il file di un file system interno o esterno al sistema e che può essere di qualunque formato.

Questo caso è un po' più complesso del precedente.

Le caratteristiche del tipo Internal sono le seguenti:

- *Ad una condizione Internal sono associati un percorso di file ed una base URL.* Il percorso di file è la stringa che sarà usata come prefisso per cercare un file generato da azioni del codice EL. La base URL sarà una stringa a cui la stringa del link risultante dalle azioni, sarà concatenata.

Ad esempio se il percorso di file di una condizione è “/tmp/docs”, la base URL è

“http://xXx.org”, la condizione è “TRUE” e il risultato delle azioni è il file “test.pdf”, il percorso di file che il gestore di link dovrà controllare sarà “/tmp/docs/test.pdf” e, se il file risulta essere esistente, il link generato sarà “http://xXx.org/test.pdf”.

- *Qualsiasi condizione del tipo Internal può avere differenti formati di file associati.* Questo è un nuovo concetto usato solo per risoluzioni del tipo Internal.

Un formato di file è semplicemente un insieme di estensioni di file che sono raggruppate sotto un identificatore. In questo modo, si può associare un identificatore di formato di file con una condizione di link. Quando la condizione è verificata il gestore combina il nome di file restituito dalle azioni EL con le estensioni di file associate per quel link.

Successivamente controlla l'esistenza di uno o più file con estensioni uguali a quelle a disposizione. In tal modo il gestore esegue la seguente composizione di stringhe per ogni estensione di file preventivamente specificata:

“Percorso di file” + “Stringa risultante dall'azione” + “Estensione di file”.

In dipendenza dal numero di file di estensioni diverse presenti nel file system, il gestore ritornerà un certo numero di link a file.

- *Quando viene invocato il gestore di link da un'espressione di linguaggio EL, se il link è risolto sarà possibile accedere ad un variabile interna speciale che contiene come valore il link risultante. Nei link di condizioni INTERNAL, tale variabile può contenere valori multipli, contenenti differenti informazioni, nel caso il gestore trovi differenti formati di file. In tal caso, è possibile utilizzare un'altra estensione che consente l'utilizzo di alcuni speciali campi di variabile contenenti i valori multipli corrispondenti e accessibili attraverso la variabile \$link quando il link è alla fine risolto*

Di seguito viene presentata una tabella con i diversi campi della variabile \$link disponibili quando una condizione del tipo INTERNAL è risolta:

<b>Nome campo</b>	<b>Descrizione del valore contenuto</b>
url	Lo stesso valore della variabile LINK: l'URL risolto
File	Il percorso assoluto al file locale a cui l'URL risolta punta
Format_id	La stringa identificatore di formato del file
Format_desc	La stringa di descrizione del formato del file definita per ogni formato di file

### 6.2.7.3 Esempio di link

In questa sezione viene presentato un esempio sull'Internal linking.

Immaginiamo di voler creare una nuova definizione di link per generare un accesso full-text ai documenti che sono archiviati su un server documentario (quindi in un file system che contenga le versioni elettroniche dei documenti). I documenti sono organizzati sistematicamente in maniera dipendente da due campi presenti nei record bibliografici: BASE, CATEGORY referenziabili internamente al codice attraverso le rispettive variabili "\$base" e "\$category".

Quando "\$base" assume il valore "CSBAREP" i file dovranno essere raggiungibili attraverso la directory "/pub/www/home/csbarep/" e verranno creati tanti link quanti sono i file in essa contenuti. I documenti sono tutti in formato HTML.

Se invece "\$base" assume il valore "CSBAPREPRINT" e "\$category" ha il valore "THESYS" o "PHOTOS" i file dovranno essere raggiungibili attraverso la directory "/archive/electronic/pub/www/home/" e verranno creati tanti link quanti sono i file presenti nella directory. In questo caso i documenti possono essere in diversi formati di file: PDF, Postscript, MS WORD.

Ovviamente, si vuole che ogni link sia creato se i file corrispondenti ai record bibliografici esistono.

Per prima cosa si crea una nuova definizione di link ad esempio chiamata FULLTEXT.

Essa riceverà tre parametri che rappresentano l'informazione necessaria per generare le due diverse possibilità di link: il tipo risolvente, il percorso del file e la base URL.

Successivamente si crea una condizione per la prima possibilità: quando il valore di \$base è "CSBAREP". Viene selezionata Internal come tipologia di link perché

si vuole effettuare un link ad un file e si vuole controllare la sua esistenza, dunque si inseriscono rispettivamente il percorso di file e la base URL con i valori corrispondenti. Si assegnano poi i tipi di formati dei file, in questo caso HTML, e si inserisce il loro criterio di archiviazione con azioni differenti.

Per l'altra possibilità: "\$base" assume il valore "CSBAPREPRINT" e "\$category" é o "THESYS" o "PHOTOS", inoltre i tipi di formato dei file assegnati saranno PDF, Postscript e MS WORD. Si procede poi allo stesso modo del caso precedente adattando le definizioni di condizioni e azioni del link ai requisiti richiesti.

Una volta finita la definizione attraverso l'interfaccia web, si possono inserire i link di questo tipo in un behavior di bibformat.

Ad esempio immaginando di aver incluso un pezzo di codice EL in un behavior che inserisce un link ai fulltext di documenti di ogni record attraverso il codice seguente:

```
link("FULLTEXT", $base, $category)  
{  
  "Fulltext: "  
  forall($link)  
  {  
    "<a href=\\" $link.url "\>" $link.format_desc "</a>"  
    separator (" - ")  
  }  
}
```

Questo codice EL includerà la stringa "Fulltext:" seguita da uno più link, separati da " - " (funzione separator), corrispondenti a tutti i documenti trovati per i valori

di variabili interne “\$base”, “\$category”, “\$id”.

Ovviamente i vari link saranno creati in base alle regole stabilite nella definizione del link FULLTEXT.

## **6.2.8 Evaluation Language (EL)**

*EL* é un linguaggio progettato dal gruppo di CDSware, per lavorare con stringhe, dotato di operatori logici e di comparazione.

Tale linguaggio viene usato per definire condizioni e azioni nelle definizioni di links, behaviors e nelle definizioni di procedure dei formats.

Nei paragrafi successivi vengono presentati brevemente i costrutti del linguaggio EL (Evaluation Language), necessario alla programmazione e configurazione degli aspetti di BibFormat.

### **6.2.8.1 Letterali**

Un *letterale* costituisce l’elemento base del linguaggio e rappresenta un valore di stringa costante racchiuso tra due doppi apici (“”).

Da ciò segue che tutto ciò che viene posto tra doppi apici rappresenta una costante letterale e, in tal caso, gli spazi e i ritorni a capo sono significativi e non ignorabili.

All’interno di un letterale é possibile esprimere il simbolo di doppio apice usando la sequenza di escape \”.

Esempi di utilizzo di letterali sono i seguenti:

**“Ciao, mondo”**

**“\“Hello, world\””**

### **6.2.8.2 Variabili**

Una *variabile* rappresenta una stringa di dati di input a cui é possibile riferirsi all'interno del linguaggio.

La sintassi di definizione di una variabile é la seguente:

**VARIABLE ::= \$ STRINGA [. STRINGA]**

Le variabili sono definite dall'utente e possono contenere campi, ossia altri valori di input raggruppati nella stessa variabile e aventi una relazione di appartenenza con essa.

Ad esempio é possibile avere una variabile che contiene informazioni sull'autore e campi quali: nome, data di nascita, luogo di nascita, ecc..

Le variabili sono case-insensitive e sono definite tramite l'utilizzo del simbolo “\$” seguito dal nome ed eventualmente da un punto ed un nome di campo nel caso ve ne siano.

La corrispondenza tra variabili e valori di input dei record viene creata attraverso il meccanismo di mappatura discusso nel 6.2.2.

Esempi di variabili sono i seguenti:

**\$autore** : in ogni parte del codice EL in cui é presente, BibFormat considererà il tag 100 associato nel record di input

**\$autore.nome** : in questo modo é possibile riferire il campo nome, corrispondente al tag 100 \$a, della variabile autore

245 \$b → **\$245.b** : il tag 245 MARC21 rappresentante il titolo principale che viene trasformato nella variabile \$245.b

### 6.2.8.3 Concatenazione di stringhe

E' possibile concatenare variabili e letterali di stringhe nel linguaggio EL in maniera implicita senza l'utilizzo di alcun specificatore, contrariamente ad esempio a quanto avviene in linguaggi come PHP dove si utilizza lo specificatore “.”.

Per effettuare la concatenazione é sufficiente mettere tali elementi l'uno dopo l'altro.

Esempi di concatenazioni sono i seguenti:

**“Autore: ” \$100.a** : effettua la concatenazione di un letterale con il contenuto della variabile rappresentante un dato del record di input

**“<b>” \$245.b “</b> - ” \$100.a** : rappresenta il titolo in grassetto (notazione HTML) seguito da un trattino e il nome dell'autore

#### 6.2.8.4 Uso delle UDFs

EL rende possibile l'utilizzo all'interno del linguaggio delle UDF, precedentemente descritte al paragrafo 6.2.2, che possono essere applicate alle stringhe.

Un'UDF è identificabile attraverso un nome di etichetta preventivamente associata ed è seguita da parametri racchiusi tra parentesi tonde. Essa restituisce una stringa come risultato, dipendente dai valori dei parametri passati.

Le UDF, come già detto, vengono definite nella sezione apposita dell'interfaccia di BibFormat.

Esse vengono poi successivamente richiamate all'interno del codice EL attraverso la seguente sintassi:

**FUNCTION ::= STRINGA ( [ PARAMETRO {,PARAMETRO} ] )**

dove:

- **STRINGA**: è una stringa indicante il nome della UDF che si intende invocare
- **PARAMETRO**: può essere una stringa rappresentante una variabile, un letterale o un'altra UDF.

Alcuni esempi di funzione sono i seguenti:

**upper(\$245.b)** : la funzione upper consente di trasformare tutti i caratteri di una stringa, in questo caso del titolo presente nel record, in maiuscolo

**copy(upper(\$100.a),“0”,“3”)** : in questo caso la funzione più interna upper agisce per prima, trasformando tutti i caratteri in maiuscolo e, successivamente, la funzione copy estrae i primi tre caratteri. L'esempio in questione mostra proprio la possibilità di poter assumere come parametro il valore ritornato da un'altra UDF.

### **6.2.8.5 Uso dei Knowledge bases**

Un'altra importante funzionalità che è possibile utilizzare nel linguaggio è l'inclusione dei KB (Knowledge Bases).

Come detto al paragrafo 6.2.5, i Knowledge Base sono un insieme di valori chiave che permettono la mappatura molti ad uno da un insieme di valori ad un unico valore standard.

I KB funzionano quindi come tabelle di traduzione per termini da normalizzare che ci si aspetta vengano utilizzati più frequentemente nei record di metadati.

La definizione dei KB viene realizzata attraverso l'apposita sezione presente nell'interfaccia web di BibFormat.

Un Kb può essere invocato all'interno di un codice in linguaggio EL attraverso la seguente sintassi:

**KB ::= kb( NOME DEL KB, VALORE CHIAVE)**

che è simile all'uso di un nome di funzione seguito da due parametri racchiusi tra parentesi,

dove:

- **kb**: è lo specificatore che permette di richiamare il KB predefinito
- **NOME DEL KB**: indica il nome del KB precedentemente definito da interfaccia web
- **VALORE CHIAVE**: indica il valore chiave da normalizzare.

Il risultato restituito da tale invocazione rappresenta il valore mappato oppure una stringa vuota se non esiste un valore chiave corrispondente ad uno tra quelli precedentemente configurati nel nome di KB specificato.

Un esempio di utilizzo é il seguente:

**kb("MESE", \$m)** : può essere usato nel caso in cui si abbiano mesi espressi con valori numerici da tradurre in nomi di mese: si dovrà definire perciò a priori un KB che mappa tutti i numeri di mese in nomi di mese.

#### **6.2.8.6 Uso dei Formats**

Come già precedentemente detto al paragrafo 6.2.4, i format sono delle etichette che permettono di raggruppare porzioni di codice EL che possono a loro volta essere utilizzate in altre espressioni del medesimo linguaggio al fine di includere tale codice.

La definizione dei format deve essere effettuata nella corrispondente sezione dell'interfaccia web di BibFormat.

La sintassi di utilizzo dei format é la seguente:

**FORMAT ::= format( NOME DEL FORMATO )**

dove:

- **format**: è lo specificatore che permette di richiamare il format predefinito
- **NOME DEL FORMATO**: è il nome rappresentante il formato definito nella corrispondente sezione di interfaccia web di BibFormat

Ad esempio possiamo pensare di dover scrivere il titolo dei record bibliografici con un certo formato: grassetto e rosso, ma tale titolo formattato deve essere utilizzato in più parti di codice EL.

E' possibile dunque definire un formato chiamato TITOLO contenete il codice:

**“<font color=\“red\”><b>” \$245.b “</b></font>”**

e riutilizzarlo per formattare i titoli dei record da stampare effettuando la chiamata seguente:

**format(“TITOLO”)**

Qualora si decida di cambiare solo la formattazione del titolo, basterà modificare semplicemente la definizione del formato TITLE.

### **6.2.8.7 Struttura “if” del linguaggio ed operatori logici e di confronto**

Il linguaggio EL possiede un'espressione *if* ed operatori logici e di comparazione integrati, per comparare espressioni e decidere l'operazione da svolgere dipendentemente dal risultato del confronto.

Gli operatori che possono essere usati nelle espressioni di confronto sono i seguenti:

1. *operatori di confronto uguale e non uguale*: (=, !=): effettuano il confronto tra due stringhe, restituiscono “True” o “False”, rispettivamente nel caso le stringhe siano identiche e nel caso non lo siano
2. *operatori logici: AND, OR e NOT*: (&&, ||, !): permettono di eseguire operazioni logiche binarie su condizioni

Tali operatori possono essere usati solo dentro espressioni che si aspettano valori logici come risultato, ossia dentro espressioni di condizione.

La sintassi del costrutto if é la seguente:

```
IF ::= if( CONDIZIONE ) { ESPRESSIONE EL } [else { ESPRESSIONE EL }  
}]
```

dove:

- **if**: specifica il costrutto per effettuare le valutazioni di tipo condizionale
- ( **CONDIZIONE** ): è un'espressione di confronto condizionale tra parentesi tonde
- { **ESPRESSIONE EL** }: è una porzione di codice EL tra parentesi graffe che sarà eseguita solo se la condizione risulta vera.
- **else { ESPRESSIONE EL }**: utilizzando else seguito da una porzione di codice tra parentesi graffe sarà possibile eseguire quest'ultima nel caso la condizione risulti falsa

Un esempio di utilizzo é il seguente:

```
format("TITLE")  
if($100.a! = "")  
{  
  "Author: "$100.a  
}
```

in tal caso il titolo sarà seguito dalla costante "Autore: " e dal nome dell'autore \$100.a solo se quest'ultimo é presente nel record di input.

#### **6.2.8.8 Uso dei Link**

Come già precedentemente discusso al paragrafo 6.2.8, BibFormat oltre che essere un interprete di linguaggio EL, é anche un risolutore di link che possiede il proprio repository delle regole per risolverli in maniera automatizzata.

Il linguaggio EL possiede una struttura speciale per richiedere al gestore di risolvere link e includerli nella versione formatta dei record bibliografici. In tal modo é possibile gestire i link in maniera facile modificandone le regole di formazione e rendendole riutilizzabili dovunque, similmente a quanto avviene per le UDF.

I link sono identificati da un'etichetta e necessitano di informazioni passate loro come parametri.

In tal modo é possibile specificare la risoluzione di un link all'interno di un'espressione EL e se risulterà esistente, BibFormat permetterà di accedere ad una variabile \$link contenete il link risolto ed altre informazioni.

Opzionalmente può essere aggiunta un'espressione else, con la stessa sintassi

vista per l'if, che sarà eseguita se il link non può essere risolto.

La sintassi da utilizzare per invocare la risoluzione di un link é la seguente:

```
LINK ::= link( ETICHETTA DEL LINK, VARIABILE) { ESPRESSIONE  
EL } [else { ESPRESSIONE EL }]
```

dove:

- **link**: è lo specificatore per invocare la risoluzione di link
- **ETICHETTA DEL LINK**: rappresenta il nome del link definito nella sezione di interfaccia web
- **VARIABILE**: rappresenta una variabile da passare alla chiamata link per determinarne la costruzione
- **{ ESPRESSIONE EL }**: rappresenta il codice EL da eseguire se il link risulta esistente tra le regole di formazione
- **else { ESPRESSIONE EL }**: permette di eseguire il codice EL tra parentesi graffe se il link risulta inesistente tra le regole di formazione

L'esempio seguente mostra il funzionamento dell'invocazione del risolutore:

```
format("TITLE") "/"  
link("AUTHOR_SEARCH", $100.a)  
{  
  "<a href=\"\"$link \"\">\" $100.a </a>\"  
}
```

in tale esempio viene applicato al titolo del record il formato TITLE e

supponendo che un tipo di link sia stato definito da interfaccia grafica attraverso l'etichetta `AUTHOR_SEARCH`, se BibFormat ha conferma dell'effettiva esistenza del link, eseguirà il codice tra parentesi graffe creando un link cliccabile all'autore che permetterà di realizzare ad esempio la ricerca per autore all'interno del repository di CDSware.

### **6.2.8.9 Costrutto Forall**

Un record bibliografico può avere più di un autore o documenti correlati di più formati che devono essere linkati. In tali casi si ha a che fare con l'applicazione di espressioni di EL su variabili o campi multivalore.

Il costrutto FORALL permette di specificare una variabile o un campo seguito da un'espressione EL tra parentesi graffe che sarà applicata ad ogni valore della variabile o del campo.

La variabile dentro il FORALL sarà riferita all'iterazione corrente.

La sintassi del costrutto é la seguente:

**FORALL ::= forall ( VARIABLE ) { espressione }**

dove:

**forall**: rappresenta lo specificatore del costrutto

**(VARIABLE)**: rappresenta la variabile di ciclo per la quale dovranno essere applicate le corrispondenti formattazioni specificate nell'espressione seguente

**{ espressione }**: rappresenta il codice EL da eseguire per tutti i valori presenti nella variabile specificata nella condizione di ciclo

Un esempio di utilizzo é il seguente:

```
format("TITLE") "/"  
forall($100.a)  
{  
  
  link("AUTHOR_SEARCH", $100.a)  
  {  
    "<a href=\"\"$link \">\"$100.a\"</a>"  
  }  
}
```

in tale esempio il costrutto forall permette di visualizzare tutti gli autori di un record bibliografico e di costruire per ognuno il link per la ricerca.

### **6.3 REALIZZAZIONE DELL'ESPORTAZIONE DEI DATI BIBLIOGRAFICI**

Il presente paragrafo ha come oggetto la realizzazione del compito assegnato, costituente peraltro la parte sperimentale dell'elaborato, che consiste nell'applicazione del modulo BibFormat, e delle sue componenti, alla risoluzione del problema inerente alla esportazione dei dati da un sistema ALEPH 500 al sistema CDSware versione 0.0.9.

Come già precedentemente accennato nell'introduzione del capitolo, l'importazione di un insieme di record bibliografici in CDSware é resa possibile fondamentalmente attraverso i moduli BibConvert e BibFormat.

La parte inerente a BinConvert che consente la trasformazione vera e propria dei record di metadati in UNIMARC, formato interno di ALEPH, al formato

MARC21 é oggetto dell'elaborato di tesi di Buttà Basilio[B3]; si faccia perciò riferimento a tale documento per qualsiasi considerazione fatta in merito a tale modulo di CDSware.

BibFormat é l'applicazione di CDSware che si occupa fundamentalmente della definizione del formato inerente alla presentazione in output dei record bibliografici all'utente finale.

I passi inerenti a tale definizione sono stati descritti in linea teorica al paragrafo 6.2.

I successivi paragrafi tratteranno le varie fasi della procedura di realizzazione del formato di rappresentazione dei record bibliografici adottato allo SBA (Servizio Bibliotecario di Ateneo) di Messina.

### **6.3.1 Mappatura dei tag per le variabili interne**

Come descritto nel paragrafo 6.2.3, questa fase si occupa della definizione dei tag che il modulo BibFormat associa alle sue variabili interne per consentire il loro riferimento nelle porzioni di codice EL utilizzabili nelle successive fasi di definizione dei behavior, delle regole dei link e dei format.

Tale associazione viene realizzata esclusivamente tramite l'utilizzo dell'apposita sezione di interfaccia web di BibFormat raggiungibile alla voce “extraction rules” presente nella barra di navigazione a sinistra della pagina web di “bibformat admin”.

Per definire l'associazione tra il tag XML MARC di un record di input e una variabile interna é necessario l'inserimento delle seguenti informazioni nelle input box presenti in fondo alla pagina web di tale sezione:

- *Type* - qui bisogna specificare un'etichetta che permette di creare il proprio

set di variabili. “DEFAULT” é l'etichetta predefinita dagli implemetatori di CDSware per mostrare i tag utilizzati più frequentemente alla biblioteca del CERN

- *Variable name* - serve a specificare il nome della variabile da associare al tag di input, ad esempio per il tag 100 si può utilizzare il nome \$100 o qualsiasi altro: in ogni caso bisognerà utilizzare sempre questo nome per referenziare il tag all'interno del codice EL
- *Field Type* - é una combo box che permette di specificare se la variabile deve essere riferita ad un “datafield”, ossia un tag che contiene dati informativi relativi alla risorsa riferita o un “controlfield”, ossia un campo di controllo specificante informazioni relative al record bibliografico in sé (ad esempio il tag 001 é un controlfield contenente il numero di identificazione del record)
- *Attribute Tag Value* - permette di specificare il valore del tag MARC21 di input; ad esempio per il campo relativo al titolo, il valore sarà 245
- *Attribute IND1 Value* - permette di specificare il primo indice del tag MARC21 [B3]. Il dato é obbligatorio solo per un campo di tipo datafield
- *Attribute IND2 Value* - permette di specificare il secondo indice del tag MARC21 [B3]. Il dato é obbligatorio solo per un campo di tipo datafield

Dopo aver definito queste informazioni, esse potranno essere confermate cliccando sul tasto “add rules”.

Successivamente, verrà visualizzata, in aggiunta a quelle già precedentemente definite dagli implementatori di CDSware, la corrispondenza tag-variabile appena creata e sarà possibile aggiungere dei sottocampi per il tag in questione attraverso il link “add subfield” presente in tale corrispondenza.

Per ogni definizione di sottocampo sarà necessario aggiungere le seguenti informazioni:

- *Subfield name* - il valore qui specificato permette di indicare la lettera di sottocampo del tag MARC21 di input
- *Attribute "label" value* - permette di specificare un'etichetta da associare al sottocampo: può corrispondere allo stesso valore assegnato al nome di sottocampo precedente

Le mappature create, relativamente all'applicazione allo SBA, per l'utilizzo successivo nelle sezioni di codice EL sono mostrate nella tabella seguente:

<b>Varname</b>	<b>Mapping Tag</b>	<b>Allow Multiple Values</b>	<b>Subfields</b>	<b>Mapping tag subfields</b>
001	<controlfield tag="001">	Yes	-	-
005	<datafield tag="005" ind1="" ind2="">	Yes	-	-
020	<datafield tag="020" ind1="" ind2="3">	Yes	\$a	<subfield code="a">
022	<datafield tag="022" ind1="" ind2="">	Yes	\$a	<subfield code="a">
100	<datafield tag="100" ind1="" ind2="">	Yes	\$a	<subfield code="a">
041	<datafield	Yes	\$a	<subfield

	tag="041" ind1="0" ind2="">			code="a">
041	<datafield tag="041" ind1="1" ind2="">	Yes	\$a	<subfield code="a">
044	<datafield tag="044" ind1="" ind2="">	Yes	\$a	<subfield code="a">
245	<datafield tag="245" ind1="1" ind2="">	Yes	\$a \$b \$c	<subfield code="a"> <subfield code="b"> <subfield code="c">
250	<datafield tag="250" ind1="" ind2="">	Yes	\$a \$b	<subfield code="a"> <subfield code="b">
362	<datafield tag="362" ind1="1" ind2="">	Yes	\$a	<subfield code="a">
260	<datafield tag="260" ind1="" ind2="">	Yes	\$a \$b \$c	<subfield code="a"> <subfield code="b"> <subfield code="c">
300	<datafield tag="300" ind1="" ind2="">	Yes	\$a \$c	<subfield code="a"> <subfield code="c">
490	<datafield tag="490" ind1="1" ind2="">	Yes	\$a \$c	<subfield code="a"> <subfield code="c">
500	<datafield	Yes	\$a	<subfield

	tag="500" ind1="" ind2="">			code="a">
760	<datafield tag="760" ind1="0" ind2="">	Yes	\$g	<subfield code="g">
246	<datafield tag="246" ind1="1" ind2="4">	Yes	\$a	<subfield code="a">
653	<datafield tag="653" ind1="0" ind2="">	Yes	\$a	<subfield code="a">
100	<datafield tag="100" ind1="1" ind2="">	Yes	\$a	<subfield code="a">
700	<datafield tag="700" ind1="1" ind2="">	Yes	\$a	<subfield code="a">
040	<datafield tag="040" ind1="" ind2="">	Yes	\$a \$e	<subfield code="a"> <subfield code="e">
980	<datafield tag="980" ind1="" ind2="">	Yes	\$a	<subfield code="a">

### 6.3.2 Definizione del comportamento (behavior) “SBA”

Come già discusso nel paragrafo 6.2.3, questa sezione si occupa della definizione dei behavior, ossia dei comportamenti, che costituiscono una sorta di contenitore

di regole che BibFormat utilizza per sapere quali operazioni di formattazione applicare ai record contenuti nel database catalogafico.

Per l'applicazione di esportazione realizzata allo SBA, si é creato un comportamento denominato "SBA" definito attraverso la sezione di interfaccia web Behaviors.

Tale sezione comporta l'inserimento di alcune informazioni in determinati campi di input e l'inserimento di un codice in linguaggio EL per la definizione delle condizioni e azioni che BibFormat valuta per la determinazione del formato applicabile.

Le informazioni, utili alla creazione di un comportamento, che é necessario inserire nelle input box sono le seguenti:

- *Output Type Name* - in tale campo é necessario specificare in "lettere maiuscole" l'etichetta che si vuole associare al proprio comportamento. Essa verrà utilizzata come parametro da passare allo script di esecuzione di BibFormat (file bibformat) per indicare il comportamento, ossia le regole generali di formattazione, che il programma dovrà applicare ad uno o più record specificati come dati di input
- *Behavior Type* - é una combo box che permette di scegliere tra due possibili tipi di comportamento da associare al formato:
  - *Input Enrich* - Crea un formato arricchito del record di input mettendo le impostazioni di formattazione alla fine del record stesso. In tal caso ogni record con tale tipo di formato se inserito nel database, avrà il proprio formato di presentazione incorporato al suo interno
  - *Normal* - crea un file di formato (ad es. fmt.xml) in cui ogni record si compone di due campi: "FMT", che contiene il formato da applicare ad ogni singolo record, e "001", che contiene il numero di

identificazione del record nel sistema. Con tale tipo, il file creato consente di inserire, tramite una procedura di uploading (descritta alla fine del capitolo), solo informazioni di formattazione e non i record bibliografici veri e propri. Ovviamente, sarà necessario in questo caso prevedere una procedura di uploading separata per i record bibliografici contenenti i metadati

- *Documentation* - é un'area di testo in cui l'utente amministratore del sistema può inserire del testo libero per descrivere il comportamento creato. E' molto utile adottare una descrizione dettagliata dei propri behavior per tutti coloro intendano riutilizzarli per la creazione di comportamenti più ricchi

Dopo aver definito queste informazioni, esse potranno essere confermate cliccando sul tasto “add output type”.

Successivamente a questa prima fase, bisognerà passare alla compilazione del codice EL relativo alle *condizioni* e alle conseguenti *azioni* del behavior SBA.

La seguente tabella, mostra il codice di entrambe le caratteristiche:

## DETAILS OF OUTPUT TYPE 'SBA'

Type: NORMAL  
Documentation: Comportamento di formattazione dei record bibliografici realizzato allo SBA di Messina

### CONDITIONS

0 \$980.a = "PICTURE"  
ACTION(0) "<record>  
<controlfield tag=\"001\"> \$001 </controlfield>  
<datafield tag=\"FMT\" ind1=\"\" ind2=\"\">  
<subfield code=\"f\">sba</subfield>  
<subfield code=\"g\">  
xml\_text(format("PICTURE\_HTML\_BRIEF"))  
</subfield>  
</datafield>  
</record>"

100 "" = ""  
ACTION(0) "<record>  
<controlfield tag=\"001\"> \$001 </controlfield>  
<datafield tag=\"FMT\" ind1=\"\" ind2=\"\">  
<subfield code=\"f\">sba</subfield>  
<subfield code=\"g\">  
xml\_text(format("DEFAULT\_HTML\_BRIEF"))  
</subfield>  
</datafield>  
</record>"

Nel caso specifico, si sono create due condizioni di valutazione:

- Condition 0 (\$980.a = "PICTURE") - tale condizione indica a BibFormat di controllare se il record di input contiene il tag MARC21 "980" (personalizzato dagli implementatori di CDSware per contenere la collezione di appartenenza del record) con valore del sottocampo \$a pari a "PICTURE". Se tale condizione è "True", BibFormat eseguirà il codice EL dell'azione ACTION(0) corrispondente. Se la condizione risulta essere

"false" andrà a valutare la condizione successiva

- Condition 100 (“”=”) - tale condizione indica a BibFormat di controllare se il record di input contiene qualsiasi tag MARC21 e, dunque, a meno che il record non contenga alcun tag, sarà sempre “True”. Una volta verificata la condizione, BibFormat eseguirà il codice rappresentante l'azione, ACTION(0), corrispondente

Il codice delle azioni rispettivamente conseguenti a Condition 0 e Condition 100 è piuttosto semplice, poiché indica a BibFormat di creare per ogni record, il cui numero identificativo è presente nella variabile “\$001”, un corrispettivo record di format contraddistinto dai tag “001” e “FMT”.

Il tag FMT è un tag appositamente utilizzato da CDSware, per indicare il comportamento da applicare al record bibliografico. Per tale tag si possono osservare il sottocampo “f”, che ha come valore il nome del comportamento da applicare, e il sottocampo “g”, che ha come valore una clausola “xml\_text” al cui interno è specificata una chiamata di procedura “format” il cui argomento rappresenta il nome del formato (definito alla sezione formats paragrafo 6.2.4) da applicare al record.

Da ciò segue che l'esecuzione della prima azione comporterà l'applicazione del formato “PICTURE\_HTML\_BRIEF” (formato HTML breve di rappresentazione delle immagini), mentre l'esecuzione della seconda, comporterà l'applicazione del formato “DEFAULT\_HTML\_BRIEF” (formato HTML breve per la rappresentazione dei documenti).

Infine, bisogna porre in evidenza i numeri di linea 0 e 100 associati rispettivamente alle due condizioni, che indicano a BibFormat quale delle due debba essere valutata per prima: ciò è importante, poiché la prima condizione è più restrittiva della seconda e se le due fossero scambiate, BibFormat non si comporterebbe nel modo corretto, perché valuterebbe sempre la condizione “” =

“”, sempre verificata, applicando il medesimo formato indiscriminatamente per tutti i record bibliografici.

### **6.3.3 Scelta del format applicabile**

Come già discusso nel paragrafo 6.2.4, questa sezione permette all'utente amministratore del sistema di creare i propri formati, che possono così essere applicati dal codice EL di un comportamento (behavior) predefinito. Nella fattispecie un “format” é un insieme di linee di codice in linguaggio EL che é possibile invocare in modo simile ad una chiamata di procedura. Questa caratteristica di BibFormat risulta essere particolarmente utile per chiunque voglia riutilizzare i codici di format già definiti (riutilizzo del software).

Tale sezione comporta l'inserimento di alcune informazioni in determinati campi di input e l'inserimento di un codice in linguaggio EL per la definizione del format specifico.

Le informazioni utili alla creazione del format sono le seguenti:

- *Format Name* - in tale campo é necessario specificare in “lettere maiuscole” il nome, ossia l’etichetta, che si vuole associare al proprio format
- *Format Documentation* - é un'area di testo in cui l'utente amministratore del sistema può inserire del testo libero per descrivere il formato creato. E' molto utile adottare una descrizione dettagliata dei propri format per tutti coloro intendano riutilizzarli per crearne di più complessi
- *EL Code* - é un'area di testo in cui inserire il codice di procedura in linguaggio EL riutilizzabile

Dopo aver inserito tali informazioni, essere potranno essere confermate cliccando sul tasto “ADD”.

Per l'applicazione di esportazione realizzata allo SBA, si sono utilizzati due comportamenti denominati “PICTURE\_HTML\_BRIEF” e “DEFAULT\_HTML\_BRIEF” predefiniti di CDSware.

Le seguenti tabelle, mostrano il codice EL dei formati utilizzati:

### **DETAILS OF FORMAT 'DEFAULT\_HTML\_BRIEF'**

Documentation Questo é il formato HTML breve di default

```
EL Code      "<strong>" format("_DEFAULT_TITLE") "</strong> "
             if(count($100.a)!="0" || count($700.a)!="0")
             {
               "/" format("_DEFAULT_AUTHORS") " "
             }
             forall($088.a)
             {
               " <small class=quicknote> [" $088.a "]"</small> "
             }
             forall($037.a)
             {
               " <small class=quicknote> [" $037.a "]"</small> "
             }
             forall($520.a)
             {
               "<br>
               <small>"
               format("_DEFAULT_ABSTRACT_FIRST_SENTENCE")
               "</small>"
             }
             forall($8564.u)
             {
               "<br>
               <small>"
```

```

format("_DEFAULT_URL")
"</small>"
}

```

## DETAILS OF FORMAT 'PICTURE\_HTML\_BRIEF'

Documentation Formato HTML breve utilizzabile per visualizzare immagini

```

EL Code      "<hr><table width=\"90%\">
              <TR>
              <TD valign=\"top\" align=\"left\">
                "<b>" format("_DEFAULT_TITLE") "</b>"
                forall($246_1.a)
                {
                  "<BR><b><i>" $246_1.a "</b></i>"
                }
                forall($260.c)
                {
                  "<BR>" $260.c
                }
                forall($520.a)
                {
                  rep_prefix("<p><i>Abstract</i>: ")
                  format("_DEFAULT_ABSTRACT_FIRST_SENTENCE")
                  "<p>"
                }
                forall($6531.a)
                {
                  rep_prefix("<BR><i>Keyword</i>: ")
                  link("KEYWORD_SEARCH",$6531.a)
                  {
                    "<a href=\"\"$link\"\">" $6531.a "</a>"
                  }
                  separator(",")
                }
                forall($037.a)
                {
                  "<BR><i>Picture number</i>: " $037.a
                }
                forall($909CP.t)
                {

```

```

        rep_prefix("<BR><I>Available picture(s)</i>:")
        xml_text($909CP.t)
    }

"</td><TD valign=\"top\" align=\"right\">"
forall($8564)
{
    if(($8564.x="icon")&&($8564.u=""))
    {
        "<P><a href=\"search?sysnb=\"$909C0.o\"\">"
        <IMG SRC="rep($8564.q," ",")"></a>"
    }
}
"<BR><font size=-2><b>CERN Geneva</b></font>"
"<BR>"link("EXT",$8564.u,$8564.z){$link}"
"</TD>"
"</TR>"
"</table>"

```

Come si può vedere dalla prima tabella, lo script di codice EL visualizza in grassetto il titolo del record bibliografico richiamato attraverso la funzione “format” che ha come argomento il format predefinito “\_DEFAULT\_TITLE”. Successivamente controlla che l'autore principale (tag 100 \$a) o gli autori secondari (tag 700 \$a) siano presenti nel record: se così, esegue la chiamata format con argomento “\_DEFAULT\_AUTHORS” per visualizzare l'autore o gli autori del record bibliografico.

Infine, esegue dei cicli iterativi “forall” per visualizzare le informazioni presenti ai seguenti tag MARC21:

- 088 \$a - é il tag contenente il numero di report aggiuntivo (additional report number) a quello indicato nel tag “037”
- 037 \$a - é il tag contenente l’identificatore interno di CDSware per identificare univocamente un record bibliografico (primary report number)

- 520 \$a - é il tag contenente l'abstract del documento a cui il record bibliografico si riferisce
- 864 \$u - é il tag contenente un URL che permette di effettuare una ricerca basata sul nome dell'autore

Il codice della seconda tabella effettua operazioni del tutto simili alla prima mettendole in una tabella, ma le informazioni che vengono visualizzate in questo caso sono relative ai seguenti tag:

- 246\_1 \$a - rappresenta una traduzione del titolo della risorsa riferita dal record
- 260 \$c - rappresenta la data di creazione della risorsa (imprint)
- 6531\_\$a - rappresenta le Keyword utilizzabili a scopi di ricerca
- 520 \$a - vale quanto detto per il tag della prima tabella
- 037 \$a - vale quanto detto per il tag della prima tabella
- 909CP \$t – rappresenta l'identificatore OAI della risorsa
- 8564\_\$u – rappresenta l'URL del fulltext del documento

#### **6.3.4 Scelta delle regole di link**

Come già discusso nel paragrafo 6.2.8, questa sezione di BibFormat consente all'amministratore di poter definire delle regole di costruzione automatica dei link basandosi sulle informazioni presenti all'interno dei record di CDSware.

Tale caratteristica rende BibFormat, oltre che un “formattatore di record”, un gestore dei link.

Relativamente a tale sezione, si é scelto, per l'applicazione di esportazione

realizzata allo SBA, di mantenere le regole di costruzione dei link predefinite di CDSware.

Le seguenti tabelle mostrano le due regole “AUTHOR\_SEARCH” e “KEYWORD\_SEARCH” di CDSware ed i relativi codici EL:

#### **DETAILS OF LINK 'AUTHOR\_SEARCH'**

<b>Evaluation Order</b>	<b>Condition EL Code</b>
0	“” = “”
<b>Actions</b>	
Action(0)	http://localhost/cgi-bin/CDSware//search?f=author&p=urlencode(\$author)

#### **DETAILS OF LINK 'KEYWORD\_SEARCH'**

<b>Evaluation Order</b>	<b>Condition EL Code</b>
0	“” = “”
<b>Actions</b>	
Action(0)	http://localhost/cgi-bin/CDSware//search?f=keyword&p=urlencode(\$keyword)

Come si può vedere dalle tabelle, le due definizioni sono del tutto simili, in quanto le condizioni di valutazione (“” = “” condizione vera per qualsiasi tag) sono identiche e BibFormat eseguirà per ciascuna regola la costruzione del link basandosi sul codice EL presente nella definizione.

Tali regole di link definiscono la possibilità di poter automatizzare la costruzione

di link di ricerca rispettivamente per l'autore di un record bibliografico (AUTHOR\_SEARCH) e per le parole chiave di un record (KEYWORD\_SEARCH). Nel primo caso l'URL di ricerca "http://localhost/cgi-bin/CDSware//search?f=author&p=" specifica "author" come argomento del parametro "f" (che sta per field) ; tale URL é seguito dalla chiamata a funzione urlencode che restituirà il nome dell'autore da ricercare.

Le stesse considerazioni possono essere fatte per la seconda regola di link "Keyword\_search".

### **6.3.5 Funzioni definibili dall'utente (UDFs)**

Come già precedentemente discusso al paragrafo 6.2.6, la sezione delle UDF permette all'amministratore di definire delle funzioni in linguaggio PHP richiamabili dalle porzioni di codice EL tramite la clausola "function" a cui verrà passato come parametro il nome della funzione PHP stessa. Tali funzioni generalmente effettuano operazioni di concatenazione e/o di trasformazione delle stringhe presenti nelle variabili interne associate ai tag dei record. Questa caratteristica di BibFormat permette quindi la cosiddetta "immersione di codice" di un linguaggio in un altro: nel caso specifico di codice PHP nel codice EL.

L'applicazione CDSware possiede numerose funzioni UDF predefinite, utili per svariati compiti di manipolazione di stringhe.

La seguente tabella mostra quelle più comunemente utilizzate:

<b>Function name</b>	<b>Params</b>	<b>Description</b>
----------------------	---------------	--------------------

COPY	Str start length	<p><b>Scopo:</b> ritorna una sottostringa a partire dalla stringa <i>str</i>, che inizia dalla posizione <i>start</i> ed è formata da un numero di caratteri pari a <i>length</i>.</p> <p><b>Esempio:</b> copy("test", "1", "2") da come risultato "es"</p>
DATE	-	<p><b>Scopo:</b> restituisce la data corrente nel seguente formato "DD:MM:YYYY:hh:mm:ss"</p>
FILE_EXISTS	url	<p><b>Scopo:</b> controlla l'esistenza del file dati il parametro <i>url</i> del sistema host che lo contiene</p>
GE	v1 v2	<p><b>Scopo:</b> ritorna 1 se <math>v1 &gt; v2</math>, "" in caso contrario</p> <p><b>Esempio:</b> ge("3", "4") dà come risultato ""</p>

Per gli scopi dell'applicazione di esportazione realizzata allo SBA, non è stato necessario definire alcuna UDF.

La definizione di un'UDF, in ogni caso, può essere realizzata dall'interfaccia della relativa sezione web attraverso l'inserimento delle seguenti informazioni nei campi di input opportunamente predisposti:

- *Function Name* - rappresenta il nome, in lettere maiuscole, da associare alla funzione da definire
- *Params* - qui vanno inseriti gli eventuali parametri che la funzione prenderà come argomenti
- *PHP Code* - è un'area di testo in cui inserire il codice PHP della funzione
- *UDF Documentation* - è un'area di testo in cui l'utente amministratore del sistema può inserire del testo libero per descrivere la funzione creata. E' molto utile adottare una descrizione dettagliata delle proprie funzioni per

tutti coloro intendano riutilizzarle per la realizzazione di altre più complesse

Dopo aver inserito tali informazioni, esse potranno essere confermate cliccando sul tasto "ADD".

La seguente tabella mostra come esempio il codice PHP della funzione "ADD" predefinita:

### **Details of UDF 'ADD'**

Documentation:	La funzione aggiunge il parametro prefix all'inizio della stringa e postfix alla fine.  sintassi: add(valore, prefisso, suffisso)  ES: add("GLI"," OPEN","ARCHIVE") costruirà la stringa: "GLI OPENARCHIVE"
PHP Code:	<pre>if(\$prefix!="")     \$value=\$prefix.\$value; if(\$postfix!="")     \$value=\$value.\$postfix; return \$value;</pre>

Tale codice é piuttosto semplice poiché verrà prima controllato che "\$prefix" non sia una stringa nulla, se é così il suo valore ("GLI" nell'esempio) verrà concatenato a quello di "\$value". Infine verrà controllato che "\$postfix" non sia nullo, se é così il suo valore ("ARCHIVE" nell'esempio) verrà concatenato a quello di \$value.

### **6.3.6 Knowledge bases (KBs)**

Come già precedentemente discusso al paragrafo 6.2.5, i Knowledge bases (KBs) rappresentano un'interessante caratteristica di CDSware con la quale si rende possibile la rappresentazione esterna in modo univoco di informazioni (dati) prestabilite.

Più precisamente tale sezione di BibFormat consente di trasformare stringhe informative differenti, presenti all'interno dei tag di un qualsiasi record di input, in un'unica stringa col medesimo significato delle precedenti, ma che ha lo scopo di dare all'utente finale una rappresentazione unica e comunemente accettata dell'informazione stessa.

Un esempio di come ciò possa essere particolarmente utile è il seguente:

consideriamo che il nome dell'autore di una qualunque risorsa bibliografica sia "Amelotti Ercole" e che tale dato informativo, ovviamente associato al tag MARC21 "100 \$a", assuma nei vari record di metadati depositati in archivio le forme: "A. Ercole", "A.E.", ecc..

Tramite i Knowledge bases è possibile stabilire un'unica rappresentazione di tale autore, ad esempio "Amelotti, Ercole", facendo in modo che quando BibFormat incontra tutte le forme precedenti, esso le trasformi nell'unica prestabilita.

Per gli scopi dell'applicazione di esportazione realizzata allo SBA, tale funzionalità di BibFormat non è stata ancora impiegata, poiché si rende necessaria una cernita di tutti gli autori presenti nell'archivio dei record, ma non si esclude che possa in futuro essere utilmente sfruttata.

La definizione di un KB, in ogni caso, può essere realizzata dall'interfaccia della

relativa sezione web attraverso l'inserimento delle seguenti informazioni nei campi di input opportunamente predisposti:

- *Knowledge base name* - serve a specificare un'etichetta, in lettere maiuscole, per identificare in maniera univoca il KB all'interno delle porzioni di codice EL: sarà possibile applicare la funzione kb ad ogni variabile contenente un valore, associato ad un tag di input, che BibFormat dovrà mappare nel corrispondente KB
- *Knowledge base table name* - rappresenta il nome della tabella che CDSware creerà per contenere le mappature dei KB. Normalmente si consiglia di non specificare alcuna tabella in tale campo, poiché l'applicazione può gestire tali tabelle in maniera automatica
- *KB Documentation* - è un'area di testo in cui l'utente amministratore del sistema può inserire del testo libero per descrivere il knowledge base creato. E' molto utile adottare una descrizione dettagliata dei propri KB per descrivere il tipo di mappatura che si intende realizzare

Dopo aver inserito tali informazioni, esse potranno essere confermate cliccando sul tasto "ADD".

La seguente tabella mostra un Knowledge Base di esempio "AUTORI\_SBA" con le relative mappature:

### Values of KB 'AUTORI\_SBA' (4 values)

Table Name	flxKB_AUTORI
Documentation	Consente la normalizzazione dei nomi di autore presenti al tag MARC21 100 \$a
A. Ercole	Amelotti, Ercole
A.E.	Amelotti, Ercole
B. Basilio	Buttà, Basilio
B.B.	Buttà, Basilio

Come si può facilmente vedere, la tabella mostra che per il KB “AUTORI\_SBA” sono state create 4 entry di mappatura di cui le prime due mappano all’autore “Amelotti, Ercole” e le seconde all'autore “Buttà, Basilio”. Si possono notare anche il nome di tabella di CDSware “flxKB\_AUTORI” creata dall'applicazione stessa per immagazzinare il KB e la stringa di testo indicante la documentazione.

#### 6.3.7 Caricamento del formato "SBA" realizzato

Dopo aver effettuato le operazioni di definizione di: mappatura dei tag, behavior, format, link e knowledge base, descritte nei paragrafi precedenti, è necessario effettuare l'ultima fase consistente nel caricamento delle impostazioni all'interno del repository delle regole di formattazione di CDSware.

Tale passo della procedura di esportazione dei record realizzata allo SBA di Messina, è strettamente correlata alla procedura di conversione, attraverso il modulo BibConvert, dal formato UNIMARC di ALEPH a quello MARC21 di CDSware documentata nell'elaborato di tesi di Buttà Basilio [B3].

Dunque, partendo dal presupposto che i record bibliografici già sottoposti a

BibConvert siano memorizzati in un file e già in formato XML MARC21, sarà possibile effettuare da linea di comando, in modalità superuser, il caricamento delle regole di formattazione secondo i seguenti passi:

*1. esecuzione di bibformat passando il parametro otype:*

```
#bibformat otype=SBA < outSBA.xml > outSBAfmt.xml
```

dove:

**bibformat** - é lo script dell'applicazione BibFormat che consente la creazione di un file che conterrà i formati di rappresentazione da associare ai record presenti nel file outSBA.xml

**otype** - rappresenta il parametro da passare a BibFormat per indicare il comportamento (behavior) di formattazione da adottare per i record in ingresso, nel caso specifico, il valore assegnato é “SBA”, comportamento la cui definizione é stata descritta nel corrispondente paragrafo

**outSBA.xml** - é il file contenente i record MARC21 di input, ottenuto dal trattamento attraverso BibConvert precedentemente menzionato. Tale file viene presentato a bibformat con un semplice reindirizzamento dell’input (“<”)

**outSBAfmt.xml** - é il file che, alla fine dell'elaborazione del modulo BibFormat, conterrà i record di formato da associare ai record bibliografici presenti nel file outSBA.xml di partenza. Tale file viene presentato a bibformat con un semplice reindirizzamento dell’output (“>”) e il file ottenuto dall'elaborazione é il seguente:

```
<record>  
<controlfield tag="001">000001189</controlfield>  
<datafield tag="FMT" ind1="" ind2="">  
  <subfield code="f">sba</subfield>
```

```

<subfield code="g">&lt;hr&gt;&lt;table width="90%"&gt;
    &lt;tr&gt;
        &lt;td valign="top"
            align="left"&gt;&lt;b&gt;&lt;/b&gt;&lt;br
&gt; c1998&lt;/td&gt; &lt;td valign="top"
            align="right"&gt;&lt;br&gt; &lt;font
            size=-2&gt;&lt;b&gt;Â©CERN
            Geneva&lt;/b&gt;&lt;/font&gt;&lt;br&gt;
            &lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;

    &lt;/subfield&gt;
&lt;/datafield&gt;
&lt;/record&gt;
&lt;record&gt;
    &lt;controlfield tag="001">000014607&lt;/controlfield&gt;
    &lt;datafield tag="FMT" ind1="" ind2="">
        &lt;subfield code="f">sba&lt;/subfield&gt;
            &lt;subfield code="g">&lt;hr&gt;&lt;table
            width="90%"&gt;
                &lt;tr&gt;
                    &lt;td valign="top"
                        align="left"&gt;&lt;b&gt;&lt;/b&gt;&lt;br&
                    &gt; 2001&lt;/td&gt; &lt;td valign="top"
                        align="right"&gt;&lt;br&gt; &lt;font
                        size=-2&gt;&lt;b&gt;Â©CERN
                        Geneva&lt;/b&gt;&lt;/font&gt;&lt;br&gt;
                        &lt;/td&gt;
                    &lt;/tr&gt;
                &lt;/table&gt;

            &lt;/subfield&gt;
        &lt;/datafield&gt;
    &lt;/record&gt;

```

2. *esecuzione dello script bibupload per il caricamento dei formati:*

**#bibupload outSBAfmt.xml**

dove:

**bibupload** - é lo script che consente il caricamento dei record bibliografici e di quelli di formato

**outSBAfmt.xml** - é il file ottenuto dall'elaborazione di cui al punto 1, che bibupload caricherà nel repository delle regole di formato consentendo a BibFormat di poter associare ad ogni record bibliografico il corrispondente record di formato.

*3. esecuzione dello script bibupload per il caricamento dei record bibliografici:*

**#bibupload outSBA.xml**

dove questa volta:

**outSBA.xml** - é il file contenente i record MARC21 di input, ottenuto dal trattamento (attraverso bibconvert) precedentemente menzionato.

L'esecuzione di bibupload si rende necessaria nel caso dell'applicazione allo SBA, in quanto il behavior "SBA", definito per essa, é di tipo "NORMAL", il quale creerà record di formato senza alcuna informazione bibliografica di arricchimento; per cui i record bibliografici devono essere caricati separatamente

Una volta effettuata la procedura di caricamento nel sistema, sarà possibile indicizzare i campi dei record bibliografici per consentire a CDSware ricerche di record basate su indici creati dalla seguente procedura da linea di comando:

**#bibwords add 1189-14607 author,keyword**

dove:

**bibwords** - é lo script che permette l'aggiornamento degli indici delle tabelle "wordsindex" e "wordsindex\_field" del database di CDSware, in

modo da consentire la ricerca per campi dei record bibliografici

**add 1189-14607** - tale opzione indica all'applicazione bibwords di aggiungere gli indici nelle tabelle "wordsindex" e "wordsindex\_field" per i record bibliografici che vanno dal numero 1189 al 14607. Tale numero di record é espresso nel tag di controllo "001" dei record bibliografici.

**author, keyword** - rappresentano gli indici (separati dalla virgola) ossia i campi dei record che dovranno essere indicizzati. Ovviamente alle parole author e keyword sono associati i corrispondenti tag MARC21 dei record di input (rispettivamente i tag "100 \$a" e "6531 \$a")

Alla fine di questa procedura sarà possibile ricercare i record bibliografici esportati attraverso l'interfaccia WebSearch di CDSware.

## BIBLIOGRAFIA

- [B3] – Buttà Basilio, *Metadati negli Open Archive: tecniche di conversione dei formati dei record nel formato MARC21 utilizzato da CDSware*, tesi di laurea in informatica, Università degli Studi di Messina, A.A. 2003-2004 (relatore Puccio L., correlatore De Robbio A.).