

# Dig *Italia*

Anno IV, Numero 1 - **2009**

Rivista del digitale nei beni culturali

ICCU-ROMA

# Una teoria fondazionale per le Biblioteche Digitali: il DELOS Digital Library Reference Model

**Leonardo Candela – Donatella Castelli\***

*Istituto di scienza e tecnologie dell'informazione "Alessandro Faedo",  
Consiglio nazionale delle ricerche di Pisa*

*Le Biblioteche Digitali sono un universo complesso. L'evoluzione che questo universo ha subito in termini di approcci, soluzioni e sistemi ha reso necessario elaborare una teoria fondazionale che permetta di unificare e sistematizzare l'insieme di conoscenze sviluppate nel settore. Il DELOS Digital Library Reference Model rappresenta un primo contributo verso questa teoria fondazionale risultante dal "collective understanding" maturato dalla comunità dei ricercatori che hanno contribuito allo sviluppo del settore. Il DELOS Digital Library Reference Model identifica un insieme di concetti e relazioni che caratterizzano il dominio delle Biblioteche Digitali e rappresenta una guida di riferimento per tutte le figure professionali coinvolte nello sviluppo ed utilizzo di tali tipologie di sistema. L'articolo introduce il Manifesto delle Biblioteche Digitali – il documento che dichiara pubblicamente l'obiettivo e la strategia per lo sviluppo della teoria fondazionale – e presenta il DELOS Reference Model – il documento che, introducendo i principali concetti e relazioni che caratterizzano le varie entità che realizzano una Biblioteca Digitale, rappresenta un primo elemento della teoria fondazionale.*

## Introduzione

Il termine "biblioteca digitale"<sup>1</sup> è stato coniato agli inizi degli anni Novanta per indicare una nuova tipologia di sistema informativo e informatico, inizialmente concepito come realizzazione elettronica o digitale di una biblioteca. Tuttavia, nel corso degli anni, tali tipologie di sistemi sono stati sviluppati in ambienti diversi, adottando una grande varietà di approcci e soluzioni. Risultato di questo processo evolutivo è l'eterogeneità di sistemi denominati "biblioteca digitale": si passa dai cataloghi bibliotecari che danno accesso agli oggetti digitali

\* Questo articolo è il frutto di una rielaborazione del volume Leonardo Candela – Donatella Castelli – Nicola Ferro – Georgia Koutrika – Carlo Meghini – Yannis Ioannidis – Pasquale Pagano – Seamus Ross – Dagobert Soergel – Maristella Agosti – Milena Dobreva – Vivi Katifori – Heiko Schuldt, *The DELOS Digital Library Reference Model: Foundations for Digital Libraries*, DELOS Network of Excellence on Digital Libraries, [S.l.]: Information society technologies, 2007.

<sup>1</sup> *Digital Libraries*, edited by Jacques Cohen, «Communications of the ACM», vol. 38, n. 4, April 1995.

corrispondenti alle descrizioni bibliografiche fino a sistemi complessi che erogano servizi avanzati per ambienti di supporto alla ricerca (Laboratori Virtuali). Questo “sovraccarico” di significati associati al termine “biblioteca digitale” è la conseguenza del fatto che la comunità coinvolta nello sviluppo del settore ha preferito adottare un approccio pragmatico e mirato alla produzione dei sistemi piuttosto che sviluppare una teoria unificante e fondazionale. L’assenza di una base comune di concetti ha avuto ricadute di carattere pratico, poiché ha penalizzato fortemente l’interoperabilità e il riuso di conoscenze, tecnologie e contenuti. Essa ha inoltre limitato la crescita degli studi nel campo delle biblioteche digitali e reso difficile, e in alcuni casi persino impossibile, sistematizzare sia le attività di valutazione e comparazione dei vari sistemi, sia l’insegnamento e lo sviluppo articolato di indirizzi di ricerca. Queste stesse difficoltà hanno ostacolato non solo il progetto e lo sviluppo dei sistemi, ma anche la ricerca di soluzioni sostenibili che abbiano come fine il riuso di conoscenze e tecniche già acquisite e il corretto approccio alle necessità delle comunità di utenti.

La ritardata definizione di una teoria unificante è anche legata alla inerente complessità del settore delle biblioteche digitali. Esso, infatti, rappresenta il punto di incontro di molti ambiti disciplinari, fra i quali quello della gestione dei dati, recupero dell’informazione, scienze bibliotecarie, documentazione, sistemi informativi, elaborazione delle immagini, intelligenza artificiale, interazione uomo-macchina, preservazione degli oggetti digitali, e non ultimo il Web. A causa di questa sua multidisciplinarietà e del diverso grado evolutivo dei settori coinvolti, è stato necessario dedicare e sviluppare interfacce e competenze volte a colmare i vuoti fra alcune delle discipline (e fra i vari ricercatori e scienziati in esse impegnati), improvvisando idee su ciò che si debba intendere per funzionalità di una biblioteca digitale e integrando soluzioni applicate in campi diversi per creare sistemi capaci di fornire tale funzionalità, essendo tali soluzioni indotte talvolta da nuovi requisiti delle biblioteche digitali. Questi requisiti sono stati identificati con una lunga indagine esplorativa, tesa soprattutto a scoprire le specifiche modalità con cui affrontare aspetti particolari della funzionalità delle biblioteche digitali. Alla fine degli anni Novanta, per esempio, il progetto ARTISTE (An Integrated Art Analysis and Navigation Environment)<sup>2</sup> si poneva l’obiettivo di come sviluppare un ambiente integrato per la ricerca e l’analisi di immagini d’arte e contenuti multimediali di simile natura; il progetto COLLATE (Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material)<sup>3</sup>, finanziato dallo stesso Programma Quadro, si è rivolgeva al trattamento di archivi storici di film, mentre il progetto ALEXANDRIA (Alexandria Digital Library)<sup>4</sup>, finanziato dalla National

<sup>2</sup> <http://www.ecs.soton.ac.uk/~km/projs/artiste>.

<sup>3</sup> <http://www.collate.de>.

<sup>4</sup> <http://www.alexandria.ucsb.edu>.

Science Foundation<sup>5</sup> nell'ambito dei programmi DLI (Digital Library Initiative) e DLI-2 si concentrava su oggetti multimediali georeferenziati. Nella maggior parte dei casi ciascun progetto ha lavorato separatamente e, in un certo senso, isolato dai restanti progetti ed iniziative aventi simili obiettivi sebbene in contesti leggermente diversi; ogni iniziativa, infatti, è partita da zero per costruire il sistema corrispondente alle particolari esigenze definite nelle specifiche di progetto. Tuttavia, guardando in retrospettiva ai risultati di tutti i progetti, si può vedere chiaramente che esiste un terreno comune a molti di loro così da poter concludere che il loro svilupparsi dal basso ha fatto emergere punti fermi sufficienti a racchiudere i risultati di tutti i loro sforzi.

La multidisciplinarietà si riflette anche nei tentativi di dare definizioni di cosa siano le biblioteche digitali, infatti ogni definizione è condizionata dall'appartenenza disciplinare del suo estensore. Fox *et al.*<sup>6</sup> osservano che l'espressione "biblioteca digitale" evoca differenti visioni, varianti da un sistema di automazione per biblioteche tradizionali fino a uno "spazio" nel quale la gente può comunicare, condividere e produrre conoscenze. Per esempio, nel suo primo incontro a San Cassiano<sup>7</sup> il Gruppo di discussione DELOS (Network of Excellence on Digital Libraries) interpretava una biblioteca digitale come un sistema capace di aprire a tutti i cittadini l'accesso a tutta la conoscenza umana, in qualsiasi momento ed in qualsiasi luogo, in modo *user friendly*, efficiente ed efficace, permettendo di superare le barriere create dalla distanza, la lingua e la cultura attraverso collegamenti Internet di diversa natura. Quindi, le biblioteche digitali sono ritenute capaci di diventare i depositi della conoscenza universale e i canali di comunicazione del futuro, il veicolo comune attraverso cui qualsiasi persona potrà trovare, discutere, valutare e accrescere qualsiasi tipo di informazione<sup>8</sup>. In modo simile, Soergel, nel suo articolo *A framework for Digital Library Research*<sup>9</sup> inizia presentando tre visioni molto diverse delle biblioteche digitali che i membri diversi di questa comunità hanno, ovvero:

<sup>5</sup> National Science Foundation – Digital Libraries, [http://www.nsf.gov/news/special\\_reports/cyber/digitallibraries.jsp](http://www.nsf.gov/news/special_reports/cyber/digitallibraries.jsp).

<sup>6</sup> Edward A. Fox – Robert M. Akscyn – Richard K. Furuta – John J. Leggett, *Digital Libraries*, «Communications of the ACM», vol. 38, Iss. 4, April 1995, p. 22-28.

<sup>7</sup> *Digital Libraries: Future Directions for a European Research Programme, Brainstorming Report, San Cassiano, Italy, 13-15 June 2001*, edited by Yannis Ioannidis, DELOS, <http://delos-noe.iei.pi.cnr.it/activities/researchforum/Brainstorming/brainstorming-report.pdf>.

<sup>8</sup> Yannis Ioannidis, *Digital libraries at a crossroads*, «International Journal of Digital Libraries», vol. 5, n. 4, August 2005, p. 255-265; Yannis Ioannidis – David Maier – Serge Abiteboul – Peter Buneman – Susan Davidson – Edward Fox – Alon Halevy – Craig Knoblock – Fausto Rabitti – Hans-Jörg Schek – Gerhard Weikum, *Digital library information-technology infrastructures*, «International Journal of Digital Libraries», vol. 5, n. 4, August 2005, p. 266-274.

<sup>9</sup> Dagobert Soergel, *A Framework for Digital Library Research*, «DLIB Magazine», vol. 8, n. 12, December 2002.

- strumenti per aiutare la ricerca, l'apprendimento e l'istruzione;
- mezzi per accedere all'informazione;
- insieme di servizi rivolti soprattutto a utenti individuali.

L'autore quindi discute le diverse concezioni prima di fonderle per definire i principi-guida per la sua visione di questo campo. D'altro canto, nel 1999 Belkin<sup>10</sup> afferma che una biblioteca digitale è una istituzione avente il compito di offrire agli utenti almeno le funzionalità di una biblioteca tradizionale nel contesto di collezioni di oggetti informativi distribuite in rete; mentre Lesk<sup>11</sup> analizza e discute singolarmente i due termini dell'espressione "biblioteca digitale", osservando che il primo comporta l'esistenza di materiale reso accessibile online da un procedimento di scansione, mentre il secondo implica l'esistenza di software per la ricerca di testi; infine conclude che il lavoro di ricerca in questo campo non tiene conto delle esigenze degli utenti. Nello stesso anno Borgman<sup>12</sup> nota che esistono due accezioni opposte dell'espressione "biblioteca digitale": i ricercatori vedono le biblioteche digitali come aggregatori di informazione a beneficio delle comunità di utenti, mentre i bibliotecari le considerano come istituzioni o servizi. Negli anni precedenti, Kuny e Cleveland<sup>13</sup> avevano discusso i quattro miti che riguardano le biblioteche digitali con l'obiettivo di invalidarli:

- Internet è la "Biblioteca Digitale";
- a un certo punto ci sarà un'unica biblioteca digitale o un'unica finestra da cui guardare le sue collezioni;
- le biblioteche digitali saranno gli strumenti per assicurare in modo più equo l'accesso al loro contenuto informativo, da qualsiasi luogo e in qualsiasi momento;
- le biblioteche digitali sono più economiche delle biblioteche "fisiche" o tradizionali.

Kuny e Cleveland concludono infine che l'esistenza delle biblioteche digitali impo-

<sup>10</sup> Nicholas J. Belkin, *Understanding and Supporting Multiple Information Seeking Behaviors in a Single Interface Framework*, in: *Proceeding of the Eighth DELOS Workshop: User Interfaces in Digital Libraries*, Biot: European Research Consortium for Informatics and Mathematics, 1999, p. 11-18, <http://www.ercim.org/publication/ws-proceedings/DELOS8/belkin.html>.

<sup>11</sup> Michael Lesk, *Expanding Digital Library Research: Media, Genre, Place and Subjects*, in: *Proceedings on the International Symposium on Digital Library 1999, ISDL'99, Tsukuba (Ibaraki), Japan, September 28-29, 1999*, Tsukuba: University of Library and Information Science, p. 51-57, [http://www.dl.slis.tsukuba.ac.jp/ISDL99/proceedings\\_ISDL99/isdl-1999-51.pdf](http://www.dl.slis.tsukuba.ac.jp/ISDL99/proceedings_ISDL99/isdl-1999-51.pdf).

<sup>12</sup> Christine L. Borgman, *What are digital libraries? Competing visions*, «Information Processing and Management», vol. 35, 1999, n. 3, p. 227-243.

<sup>13</sup> Terry Kuny – Gary Cleveland, *The Digital Library: Myths and Challenges*, in: *62nd IFLA General Conference, Beijing, 25-31 August 1996*, <http://www.ifla.org.sg/IV/ifla62/62-kuny.pdf>.

ne di reinventare il ruolo del bibliotecario e il modello della biblioteca.

Malgrado la varietà delle angolazioni con cui si può considerare oggi una biblioteca digitale, la sua concezione si è evoluta sostanzialmente da quella originaria, che vedeva la biblioteca digitale come un sistema per permettere l'accesso a libri digitalizzati e documenti testuali di simile natura. La Rete di eccellenza DELOS per le Biblioteche Digitali<sup>14</sup> interpretava le biblioteche digitali quali strumenti al centro di ogni attività intellettuale, privi di limiti o barriere – concettuali, logiche, temporali o personali – alla loro capacità di trattare le informazioni. Così le biblioteche digitali sono passate gradualmente da essere concepite come sistemi basati sul contenuto, e quindi capaci semplicemente di organizzare e dare accesso a particolari collezioni di dati e informazioni, a sistemi orientati agli utenti capaci di fornire un ambiente personalizzato di supporto alle varie attività legate alla produzione di nuova conoscenza. Il ruolo principale di un biblioteca digitale perciò è passato da quello di sistema per permettere il recupero delle informazioni staticamente memorizzate a quello di “laboratorio virtuale”, ossia strumento per facilitare la comunicazione, la collaborazione ed altre forme di interazione fra scienziati, ricercatori, e tante altre tipologie di utenti, su temi pertinenti le informazioni che esse memorizzano. In parallelo a questa evoluzione, esse sono passate da trattare dati testuali localmente centralizzati ad offrire la possibilità di trattare e processare in modo integrato collezioni di oggetti multimediali distribuiti, dati provenienti da sensori, apparecchiature mobili o strumenti di calcolo di grande diffusione.

A causa di tutte queste diverse sfaccettature la nozione di “biblioteca digitale” è complessa, variegata, e difficile da esprimersi con una semplice definizione. Il *Manifesto delle Biblioteche Digitali* è stato introdotto al fine di catturarne gli aspetti essenziali. L'obiettivo di questo documento è stabilire i fondamenti e identificare i concetti basilari del mondo delle biblioteche digitali, così da facilitare l'integrazione dei vari filoni di ricerca e proporre una guida per lo sviluppo di sistemi appropriati.

## **Il Manifesto delle Biblioteche Digitali**

Il *Digital Library Manifesto*<sup>15</sup> si basa sulle conoscenze sviluppate da gruppi di ricerca europei sulle tematiche delle biblioteche digitali, inclusi i gruppi che fanno parte della Rete DELOS, e sui risultati degli eventi DELOS dedicati a tali tematiche (San Cassiano, 2001; Corvara, 2004; Frascati, 2006).

La prima macro dichiarazione che gli estensori del *Manifesto* effettuano per poter discriminare tra le diverse tipologie di sistemi per biblioteche digitali consiste nel distinguere tre concetti di “sistema”.

<sup>14</sup> DELOS Network of Excellence on Digital Libraries, <http://www.delos.info>.

<sup>15</sup> Leonardo Candela – Donatella Castelli – Georgia Koutrika – Yannis Ioannidis – Pasquale Pagano – Seamus Ross – Hans-Jörg Schek – Heiko Schuldt, *The Digital Library Manifesto*, DELOS Network of Excellence on Digital Libraries, Pisa: Information society technologies, 2006.

La Figura 1 presenta queste tre diverse nozioni di “sistema” risultanti da questo processo evolutivo.

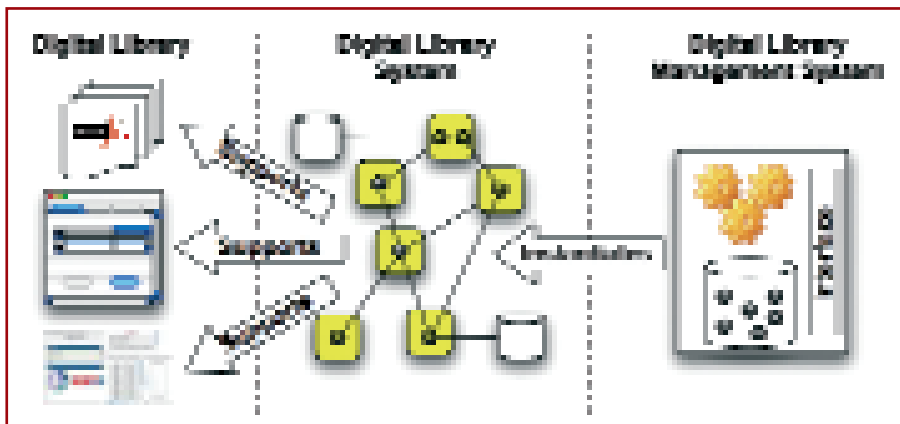


Figura 1. DL, DLS e DLMS: i tre “sistemi”

Le tre nozioni di sistema sono spesso confuse e usate in modo indifferenziato nella letteratura, e questa mancanza di precisione terminologica ha prodotto una pleora di oggetti eterogenei e contribuito a rendere difficile la descrizione, la comprensione e lo sviluppo dei sistemi per biblioteche digitali. Come mostrato dalla Figura 1, ciascuno dei tre sistemi gioca un ruolo centrale ma distinto nel processo di sviluppo di una biblioteca digitale. Le caratteristiche di ciascun sistema sono individuate con l'aiuto delle definizioni che seguono<sup>16</sup>:

- DL (Digital Library, Biblioteca Digitale) è un'organizzazione (eventualmente virtuale) che in modo organico raccoglie, gestisce e preserva per il lungo periodo un ricco contenuto digitale (Content) ed offre alle sue comunità di utenti (User) le funzionalità (Functionality) specializzate a trattare tale contenuto con livelli di qualità (Quality) misurabile e secondo regole (Policy) definite;
- DLS (Digital Library System, Sistema di Biblioteca Digitale) è un sistema software, basato su un'architettura (Architecture) ben definita (eventualmente distribuita), che fornisce tutte le funzionalità richieste da una specifica DL. Gli utenti interagiscono con la DL attraverso il corrispondente DLS;
- DLMS (Digital Library Management System, Sistema di Gestione di Biblioteche Digitali) è un sistema software che fornisce l'infrastruttura software adatta a:

<sup>16</sup> Per rendere più comprensibili i riferimenti alle figure inserite nel testo, verranno usati nel seguito gli acronimi derivati dai nomi in lingua inglese. Nel seguito tali acronimi verranno usati per indicare gli specifici sistemi da essi identificati, mentre verrà usato il termine “biblioteche digitali” per far riferimento ai sistemi fino ad oggi oggetto di ricerca da parte dell'omonimo settore.

- produrre e amministrare un DLS fornito di tutte le funzionalità considerate fondamentali per qualsiasi DL;
- incorporare funzionalità aggiuntive, più accurate, specializzate o avanzate.

Mentre esempi di DL e DLS possono facilmente essere incontrati nella vita quotidiana – un qualsiasi utente che usi il contenuto digitale di una qualsiasi DL probabilmente troverà nell’interfaccia che sta usando il nome del “suo” DLS. Appare tuttavia opportuno aggiungere anche alcune informazioni sui DLMS. A questo scopo può essere utile ricorrere ad una approssimativa analogia tra il mondo delle biblioteche digitali e quello dei sistemi di gestione di basi di dati (Database Management System, DBMS). Un DBMS – ad esempio Oracle o DB2 – può essere paragonato ad un DLMS nel suo offrire servizi generali per la gestione dei dati. Una specifica installazione di un DBMS, insieme a tutto il suo software applicativo, può essere paragonato a un DLS. Infine, una DL si può mettere in relazione al “sistema informativo” realizzato con un DBMS, costituito dal software, dai suoi dati, dai suoi utenti. Quindi un DLMS appartiene alla classe del “software di sistema” (o “software di base”). Similmente ad altri sistemi della stessa classe, è provvisto di un ambiente di generazione di nuovo software. In questo suo ambiente un DLMS mette a disposizione meccanismi usabili come piattaforma per creare sistemi DLS. Il concetto di DL fa riferimento ad un sistema astratto consistente di componenti sia fisici che virtuali, i concetti di DLS e DLMS si riferiscono invece a sistemi software concreti. Ogni DL è resa funzionante da un unico DLS, eventualmente costituito da più DLS interconnessi. Tutti i DLS, invece, sono basati su un piccolo insieme di DLMS. Esempi di DLMS sono MARIAN<sup>17</sup>, il DelosDLMS<sup>18</sup> e gCube con il suo DL Generator<sup>19</sup>.

Una volta introdotte queste tre nozioni di sistema esistenti nel mondo delle biblioteche digitali, si possono identificare i concetti principali che li caratterizzano. Questi sono descritti nelle sezioni seguenti.

### **Il mondo delle biblioteche digitali: concetti principali**

Malgrado la grande varietà di biblioteche digitali esistenti, i concetti considerati fondamentali – ed effettivamente identificabili in quasi tutte le tipologie di sistemi

<sup>17</sup> Marcos A. Gonçalves, *Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Model for Digital Library Framework and Its Applications*, PhD thesis, Virginia Polytechnic Institute and State University, November 2004.

<sup>18</sup> Yannis Ioannidis – Diego Milano – Hans-Jörg Schek – Heiko Schuldt, *DelosDLMS From the DELOS vision to the implementation of a future digital library management system*, «International Journal on Digital Libraries», vol. 9, n. 2, November 2008, p. 101-114.

<sup>19</sup> Massimiliano Assante – Leonardo Candela – Donatella Castelli – Luca Frosini – Lucio Lelii – Paolo Manghi – Andrea Manzi – Pasquale Pagano – Manuele Simi, *An Extensible Virtual Digital Libraries Generator*, in: *Proceedings of 12th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2008, Aarhus, Denmark, 14-19 September 2008*, Berlin: Springer, c2008.



attualmente operative – appartengono ad un nucleo molto ristretto. Questo nucleo di concetti può servire allo studioso che voglia comprendere questa area disciplinare, al progettista e allo sviluppatore che voglia realizzare una DL, ed a ogni fornitore di contenuti digitali che voglia “esporre” tali contenuti con le tecnologie utilizzate in questo campo.

I concetti (o aree concettuali) che caratterizzano il mondo delle biblioteche digitali sono sei e sono riportati in Figura 2. Cinque di essi compaiono nella definizione di una biblioteca digitale (DL) introdotta nel paragrafo precedente: contenuto digitale (Content), utenti (User), funzionalità (Functionality), livelli di qualità (Quality) e regole (Policy); il sesto compare nella definizione di DLS: architettura (Architecture).

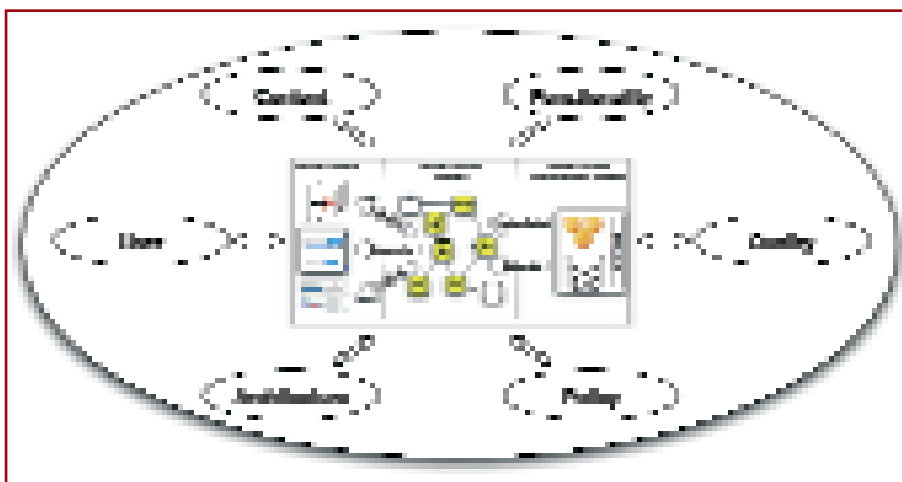


Figura 2. Il mondo delle biblioteche digitali: concetti principali

Il concetto Content fa riferimento a tutti i dati e informazioni che una biblioteca digitale, intesa nella sua interezza, gestisce e mette a disposizione dei suoi utenti. Content è un concetto generale usato per comprendere tutte le forme di oggetti informativi che ogni sistema per biblioteca digitale colleziona, organizza e rende disponibili, incluse annotazioni e metadati, cioè dati ben identificabili, capaci di rappresentare sinteticamente ogni tipo di risorsa informativa.

Il concetto User comprende tutti gli attori – siano essi umani o macchine – capaci di interagire con una biblioteca digitale intesa nella sua interezza. Infatti il ruolo principale assegnato a questi sistemi è quello di connettere gli attori con l’informazione (Content) e offrire loro un ambiente per il consumo dell’informazione al fine di generare nuova conoscenza. Il concetto di User copre tutte le nozioni relative alla rappresentazione e alla manipolazione delle entità-attore all’interno di

tutti i sistemi di biblioteca digitale (DL, DLS e DLMS), e quindi comprende entità quali i diritti di proprietà intellettuale da rispettare nella gestione delle informazioni, o i “profili” degli attori che individuano caratteristiche adatte a personalizzare il comportamento del sistema biblioteca digitale nei loro confronti o a rendere possibile la collaborazione fra attori con profili simili.

Il concetto Functionality si riferisce a tutti i servizi che una biblioteca digitale, nella sua interezza, offre alle diverse tipologie di utenti, siano essi individui o classi di utenti. Mentre ci si aspetta, in generale, che una biblioteca digitale sia ricca di servizi, il loro insieme minimo deve includere gli aspetti relativi alla registrazione di nuovi oggetti informativi, la ricerca e lo scorrimento (*browse*). Oltre a ciò, il sistema biblioteca digitale ha il compito di gestire le sue funzioni in modo tale da accogliere le particolari esigenze della comunità dei suoi utenti e/o soddisfare i requisiti specifici del contenuto che è chiamato a gestire.

Il concetto Quality rappresenta i parametri che possono essere usati per caratterizzare e valutare il contenuto e il comportamento di un sistema biblioteca digitale in ogni sua componente. La Quality può essere riferita non solo ai parametri di qualsiasi classe di contenuto o funzionalità, ma anche a quelli di specifici servizi o oggetti informativi. Alcuni di questi parametri sono oggettivi per loro natura, e perciò il loro valore può essere misurato automaticamente, mentre altri sono necessariamente soggettivi e possono essere valutati solo con un giudizio da parte di gruppi di utenti.

Il concetto Policy fa riferimento all’insieme, o agli insiemi, di regole, condizioni, norme, disposizioni o accordi che governano le interazioni tra il sistema biblioteca digitale e i suoi utenti, sia reali che virtuali. Esempi di Policy sono le norme di comportamento degli utenti, il trattamento dei diritti di autore, il rispetto della privacy, le tariffe, le modalità di consegna dei documenti. Le Policy appartengono a classi differenti, e non tutte le classi di Policy compaiono in una biblioteca digitale o nell’organizzazione responsabile della sua gestione.

Il concetto Architecture si riferisce alla parte più sistemica delle biblioteche digitali, cioè al DLS e al DLMS, e rappresenta il risultato della mappatura della funzionalità e del contenuto offerti dalla DL su componenti hardware e software.

Questi sei concetti catturano caratteristiche ortogonali e complementari dei sistemi per biblioteche digitali. Tuttavia, sono caratterizzati da altre relazioni che sono riportate in Figura 3. Tre di tali concetti sono indipendenti, cioè la loro esistenza non è legata all’esistenza di una DL. Essi sono: Architecture, che si riferisce al progetto tecnologico su cui è basato il DLS; User, che rappresenta gli attori umani o i componenti hardware che interagiscono con la DL o il DLS, e infine Content, che rappresenta il materiale informativo trattato dalla DL e dal DLS. Al di sopra di questi si pone il concetto di Functionality, che rappresenta le procedure, le azioni e le interazioni che come primo obiettivo vogliono mettere in comunicazione User e Content. Infine, le operazioni del sistema biblioteca digitale e l’attivazione delle sue Functionality sono basate su Policy e aspirano a raggiungere una certa Quality.

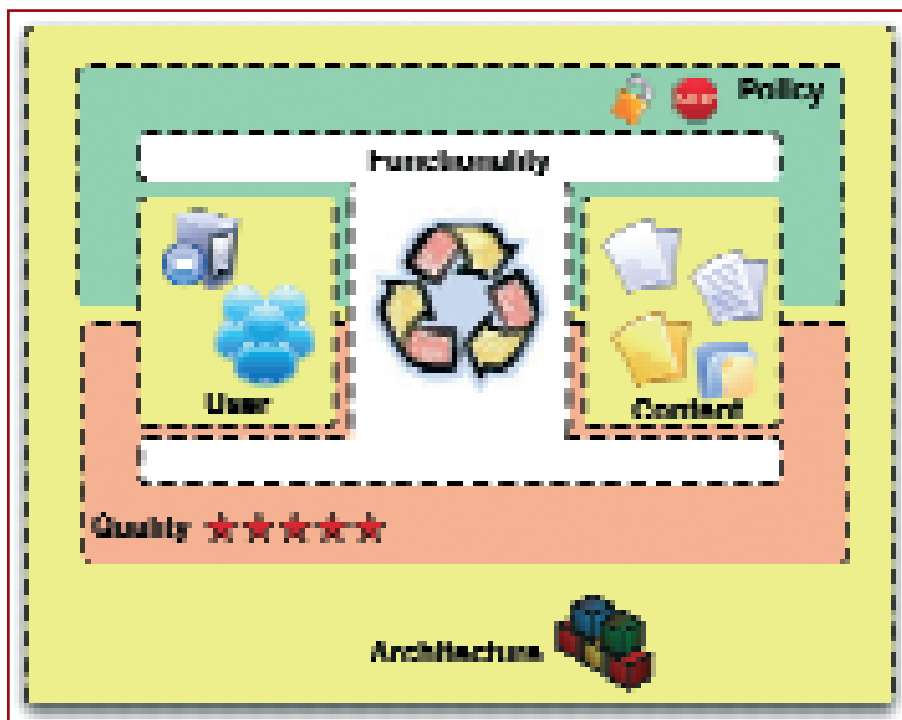


Figura 3. Il mondo delle biblioteche digitali: relazioni tra concetti principali

Dopo aver definito i sistemi ed i concetti che caratterizzano il mondo delle biblioteche digitali, il *Manifesto* introduce le classi di attori o meglio i principali ruoli che entrano in gioco in una biblioteca digitale impiegata nella sua interezza, ovvero usare la DL e mantenere operativo il DLS che la supporta attraverso le funzionalità del DLMS.

### Il mondo delle biblioteche digitali: i ruoli principali degli attori

Gli attori interagiscono con un sistema biblioteca digitale nella sua interezza possono giocare uno dei seguenti quattro ruoli principali, diversi e complementari tra loro: utente finale (DL End-User), progettista (DL Designer), amministratore di sistema (DLS Administrator) e sviluppatore di applicazione (DL Application Developers). Come mostrato in Figura 4, ciascun ruolo è associato principalmente a uno dei tre “sistemi” rappresentati nella struttura a tre livelli già discussa. Mentre gli utenti finali interagiscono principalmente con la DL, gli attori degli altri ruoli interagiscono principalmente con il DLMS.

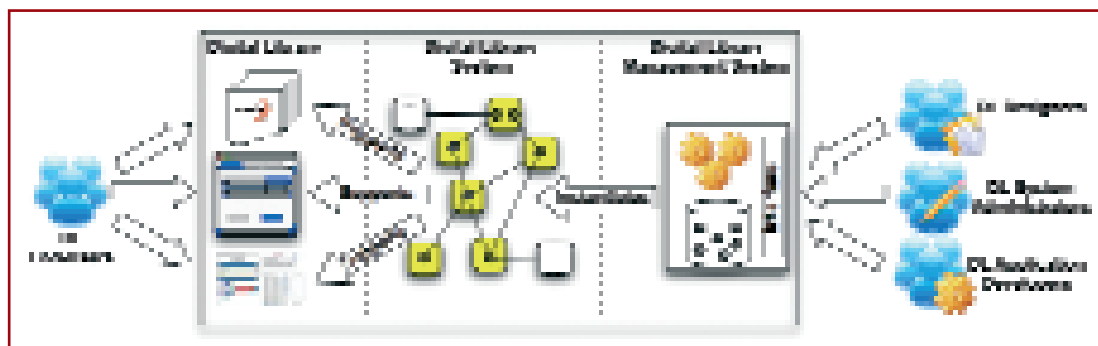


Figura 4. I ruoli principali degli attori rispetto alla struttura a tre livelli

Gli utenti finali (End-user) usano le funzionalità della DL per creare, usare e gestire il contenuto della DL stessa. Essi percepiscono la DL come un'entità affidabile che soddisfa i loro bisogni. Il comportamento ed i risultati della DL dipendono dallo "stato" della DL al momento in cui certe sue funzionalità sono attivate, intendendosi per "stato" la condizione delle sue risorse che, come già visto, sono costituite dalle collezioni di oggetti informativi gestiti dalla DL, dall'insieme degli utenti autorizzati, dall'insieme delle sue funzionalità e dall'insieme delle sue regole. Questo stato cambia durante il ciclo di vita della DL come conseguenza dell'uso della stessa da parte di questa tipologia di utenti. Tra gli utenti finali si possono identificare tre ruoli distinti: il creatore di contenuti (Content Creator), il consumatore di contenuti (Content Consumer) e il bibliotecario (Librarian).

Il creatore di contenuti è chi alimenta l'insieme delle risorse informative di una DL – principalmente il pool di oggetti informativi – a cui possono accedere altri utenti finali. Questa creazione di contenuti è svolta :

- attraverso le funzionalità (Functionality) della DL;
- in conformità delle regole (Policy) vigenti nella DL;
- in accordo ai livelli di qualità (Quality) che la DL è impegnata a garantire.

Il consumatore di contenuti è chi acquisisce/usufruisce del contenuto della DL. In realtà questi utenti utilizzano tutte le risorse che la DL mette a disposizione; infatti essi accedono ai contenuti:

- attraverso le funzionalità (Functionality) offerte dalla DL;
- in conformità delle regole (Policy) vigenti nella DL;
- in accordo ai livelli di qualità (Quality) che la DL è impegnata a garantire.

Il bibliotecario<sup>20</sup> è l'utente finale incaricato di curare i contenuti (Content) della DL, ad esempio definire le regole (Policy) che ne governano l'uso.

I progettisti (DL Designer) usano la loro conoscenza del dominio semantico che la DL è chiamata a servire per definire, personalizzare e mantenere la DL oggetto del loro lavoro così da allinearla ai bisogni di informazioni e funzionalità dei potenziali utenti finali. Per fare questo, essi interagiscono con il DLMS fornendo i parametri di configurazione che caratterizzano le funzionalità (Functionality) e i contenuti (Content) della DL. Questi aspetti includono il modo in cui sono mostrati i risultati delle varie funzionalità, il linguaggio (o i linguaggi) di interrogazione, le informazioni da includere nel profilo degli utenti, il modello dei documenti e dei dati, la lista degli archivi e collezioni da gestire, le ontologie, schemi di classificazione, liste di controllo<sup>21</sup>, ecc. I valori di questi parametri prefigurano il modo in cui tale DL si presenterà ai suoi utenti finali, perché essi determinano il particolare DLS che rende operativa la DL stessa. Naturalmente tali parametri possono essere modificati durante il ciclo di vita della DL per rispondere al mutare delle esigenze degli utenti finali rispetto a una o più delle caratteristiche della DL.

Gli amministratori di sistema (DLS Administrator) selezionano i componenti software necessari a realizzare uno specifico DLS. La scelta di tali componenti è guidata dalle aspettative degli utenti finali e dei progettisti di una DL, ma anche dai requisiti imposti dalla disponibilità di risorse, ad esempio dalla tipologia delle risorse hardware (server) da utilizzarsi per ospitare i servizi del DLS. Gli amministratori di sistema interagiscono con il DLMS per configurare l'architettura del DLS, specificando il valore dei parametri riguardanti, ad esempio, l'allocazione dei componenti software rispetto alle componenti hardware. Il loro compito è identificare la configurazione dell'architettura più appropriata a che il DLS assicuri il livello di qualità (Quality) atteso. Il valore dei parametri di configurazione dell'architettura possono essere cambiati nel ciclo di vita della DL.

Ogni cambiamento di questi parametri può risultare nella fornitura di diverse funzionalità della DL e/o di diversa qualità dei suoi servizi.

Gli sviluppatori di applicazioni (DL Application Developer) creano i componenti software che costituiranno il DLS, assicurando la presenza di tutte le funzionalità necessarie e il loro appropriato livello di qualità. In questa classe sono compresi i progettisti del software. Infatti, sebbene la creazione di una DL sia basata sulla presenza di un DLMS che offre un insieme di tecnologie, l'adattamento di queste o anche lo sviluppo di nuove tecnologie può essere richiesto per soddisfare i bisogni di una specifica comunità di utenti.

<sup>20</sup> Il significato di questo termine non va confuso con le altre accezioni/funzioni che si assegnano al termine bibliotecario. Come sarà chiarito nel seguito i bibliotecari giocano un ruolo più ampio che quello di un semplice utente finale.

<sup>21</sup> Traduciamo con "liste di controllo" il termine "Authority files" con il quale si indicano archivi di elementi della descrizione bibliografica – generalmente titoli o autori – con i quali si può controllare il corretto uso di tali elementi nelle citazioni.

Dopo aver analizzato i ruoli riconosciuti dal *Manifesto*, si può essere sorpresi del fatto che il bibliotecario sia menzionato come un utente finale. In realtà, oggi il bibliotecario<sup>22</sup> può essere partecipe di molti dei ruoli che sono stati descritti<sup>23</sup>.

Poiché l'utente finale esegue funzionalità per fornire, utilizzare e trattare i contenuti della DL, il bibliotecario è in questa categoria per i compiti che esso svolge come catalogatore e curatore, e per il suo ruolo di interfaccia con gli utenti della DL. Questo speciale ruolo è svolto attraverso la DL stessa, poiché nel mondo delle biblioteche digitali non esiste un luogo fisico che rappresenti la DL.

Poiché il progettista deve avere conoscenza del dominio semantico dell'applicazione per esser capace di definire, configurare e mantenere la DL, il suo lavoro deve essere affiancato da quello del Digital Librarian, cioè dal bibliotecario responsabile delle regole di funzionamento della biblioteca.

Infine, l'amministratore di sistema deve essere affiancato dal System Librarian, un bibliotecario con competenze tecniche che lo rendono capace di gestire il software che supporta il funzionamento della DL.

Quindi, tutte le "incarnazioni" delle funzioni che i bibliotecari sono chiamati a svolgere oggi sono coperte dai ruoli previsti nel *Manifesto*.

I quattro ruoli appena descritti comprendono l'intero spettro degli attori che interagiscono con il mondo delle biblioteche digitali. Le differenti visioni che gli attori, nei differenti ruoli, hanno di questo universo, cioè i loro diversi "modelli" di biblioteca digitale, sono legati da una relazione di natura gerarchica, risultante dal fatto che gli utenti finali (DL End-users) interagiscono con la DL, mentre i progettisti (DL Designers), gli amministratori di sistema (DLS Administrators) e gli sviluppatori di applicazioni (DL Application Developers) operano sul corrispondente DLS (attraverso la mediazione del DLMS) e, come conseguenza, sulla DL medesima. Questa relazione di inclusione assicura che i vari attori possano condividere linguaggio e conoscenze durante il loro lavoro, necessariamente di natura cooperativa.

Ad esempio, un utente finale può esprimere requisiti in termini di modello di DL, lo stesso a cui fa riferimento il progettista che può quindi comprendere tali requisiti e definire la DL in modo da soddisfarli.

Questa relazione di inclusione dei modelli che ciascun utente percepisce è presentata in Figura 5.

Come risultato ultimo e dopo aver presentato i principali aspetti che caratterizzano il mondo delle biblioteche digitali, il *Manifesto* provvede a definire un modello coerente che porta allo sviluppo dei sistemi.

<sup>22</sup> Il termine "bibliotecario" (non in corsivo) è utilizzato nel resto del testo per indicare la classe di attori che svolgono le funzionalità legate alla "gestione" della biblioteca digitale e non deve essere confuso con ruolo *librarian* definito in precedenza.

<sup>23</sup> Leonardo Candela – Donatella Castelli – Pasquale Pagano, *On-demand Virtual Research Environments and the Changing Roles of Librarians*, «Library Hi Tech», vol. 27 Iss. 2, 2009, p. 239-251.

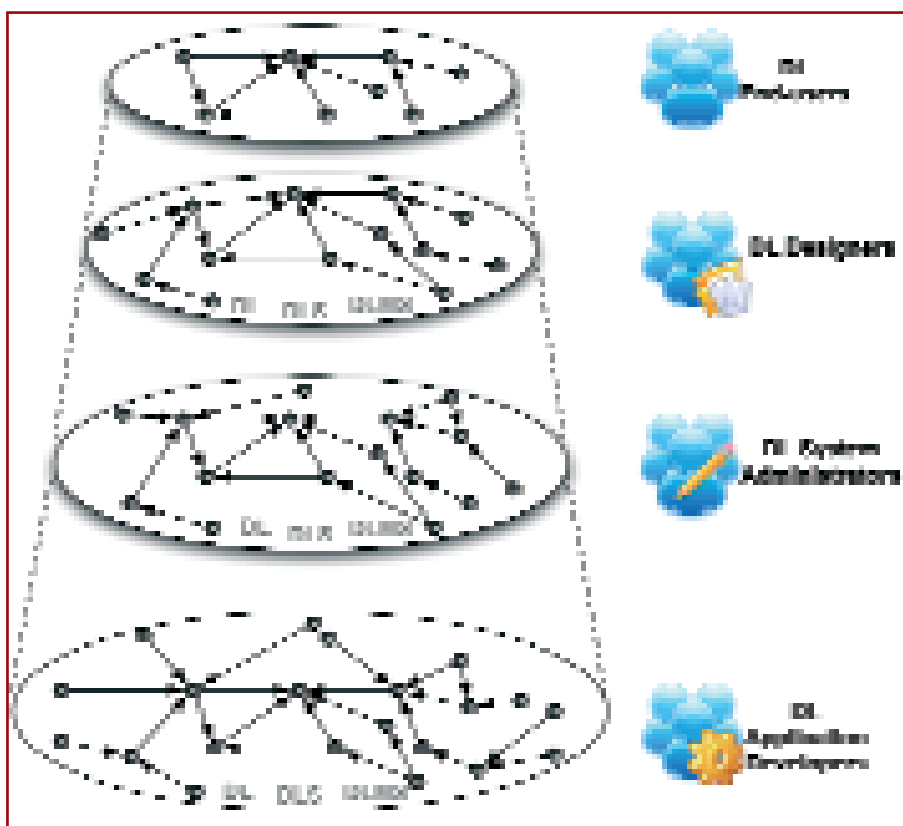


Figura 5. Gerarchia delle viste/prospettive utente

### Il mondo delle biblioteche digitali: un modello di sviluppo

Come abbiamo visto, il mondo delle biblioteche digitali è complesso, ed è perciò difficile individuare un unico modello, completamente sviluppato, capace di esprimere accuratamente tutti gli aspetti necessari a rappresentare questo mondo nei diversi scenari. Uno degli scenari nei quali l'attività di modellazione è particolarmente importante è il percorso che porta allo sviluppo di sistemi concreti.

Questo scenario è molto ampio, perché la "cattura" delle peculiarità di ciascuna entità ad un livello di dettaglio tale da permettere allo sviluppatore di implementare tale entità richiede la capacità di "catturare" l'insieme completo degli aspetti che caratterizzano quella entità; questa modellazione capillare, in particolare l'insieme delle peculiarità catturate dal modello, potrebbe poi essere riusata in molti altri domini applicativi, per esempio nell'insegnamento. Un tale modello, tuttavia, può essere di difficile uso se non viene progettato in modo appropriato, cioè conformato in modo da adattarsi alle specifiche esigenze per le quali è costruito.

Per queste ragioni nel *Manifesto* si introduce una struttura per il modello di svi-

luppo per il mondo delle biblioteche digitali (vedi Figura 6). I molteplici elementi che lo costituiscono possono essere meglio compresi e rappresentati nei loro dettagli se si fa ricorso a contesti che permettano differenti livelli di astrazione. In particolare, i contesti di riferimento sono:

- Modello di riferimento (Reference Model). Come affermato nel modello di riferimento per i sistemi software con architetture orientate ai servizi, «un modello di riferimento consiste in un insieme minimo di concetti unificanti, di assiomi e relazioni riguardanti un particolare dominio; esso vale indipendentemente da specifici standard, tecnologie, implementazioni o altri dettagli concreti»<sup>24</sup>. Le biblioteche digitali hanno bisogno di un loro modello di riferimento per trasformare la diversità delle attuali concezioni in un tutto coeso e fondato, per offrire meccanismi di comparazione fra i diversi sistemi, per creare una base comune di comunicazione all'interno della comunità degli addetti, per aprire nuovi sviluppi.
- Architettura di riferimento (Reference Architecture). L'architettura di riferimento è uno schema di progetto che indica una soluzione astratta che implementa i concetti e le relazioni identificate nel modello di riferimento. Può esistere più di una architettura di riferimento indirizzata al progetto di sistemi per biblioteche digitali (siano DLS o DLMS) costruiti in conformità dello stesso modello di riferimento. Ad esempio, potremmo avere una architettura di riferimento per sistemi DLS con cui poter realizzare DL attraverso la federazione di risorse locali e molteplici organizzazioni, ed una diversa architettura di riferimento per realizzare DL personali o applicazioni specializzate.
- Architettura Concreta (Concrete Architecture). A questo livello si realizza l'architettura di riferimento; tale realizzazione è ottenuta sostituendo i meccanismi individuati nell'architettura di riferimento con standard, specifiche e tecnologie concrete. Per esempio, una architettura concreta può specificare che l'ambiente di supporto alle comunicazioni tra i componenti software è basata su CORBA (Common Object Request Broker Architecture) piuttosto che su implementazioni degli standard per Web Services.

La relazione di questi tre elementi è mostrata in Figura 6<sup>25</sup>. A livello più alto si colloca l'elemento più astratto, il modello di riferimento, che fa da guida alla più specifica architettura di riferimento e alla sottostante architettura concreta. Questi tre elementi, che dovrebbero condizionare lo sviluppo e l'implementa-

<sup>24</sup> C. Matthew MacKenzie – Ken Laskey – Francis McCabe – Peter F Brown – Rebekah Metz, *Reference Model for Service Oriented Architecture 1.0., OASIS Standard*, October 2006, <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>.

<sup>25</sup> La figura è stata suggerita dall'articolo MacKenzie, *Reference Model* cit.



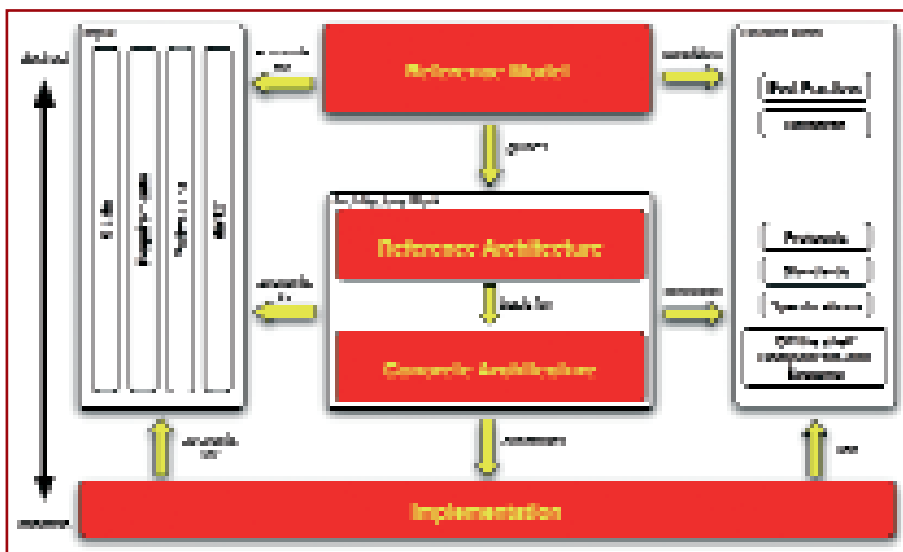


Figura 6. Un modello di sviluppo per sistemi di biblioteche digitali

zione di qualsiasi sistema reale, sono il risultato di un processo di astrazione che ha tenuto conto sia degli obiettivi, dei requisiti, delle motivazioni, e, in generale, del mercato, come indicato nel lato sinistro della figura, sia delle pratiche e della ricerca, come mostrato sul lato destro della stessa. Quando le indicazioni di questo contesto di riferimento sono adottate e seguite, i sistemi che ne risultano saranno ampiamente compatibili l'uno con l'altro. L'interoperabilità che conseguentemente ne deriverà aprirà nuovi e significativi orizzonti alla ricerca nel campo.

## Il DELOS Reference Model

Un modello di riferimento (Reference Model) è una struttura concettuale utile a rappresentare le entità significative di un certo universo insieme alle loro relazioni. Per tale rappresentazione vengono normalmente usati linguaggi di modellazione, quali ad esempio, Entity-Relationships<sup>26</sup>, UML (Unified Modeling Language)<sup>27</sup>, Description Logic<sup>28</sup>.

Per la rappresentazione del Reference Model per le biblioteche digitali, si sono

<sup>26</sup> Peter P. Chen, *The Entity-Relationship Model: Toward a Unified View of Data*, «ACM Transactions on Database Systems», vol.1, n.1, March 1976, p. 9-36.

<sup>27</sup> Grady Booch – James Rumbaugh – Ivar Jacobson, *The Unified Modeling Language User Guide*, Upper Saddle River (NJ): Addison Wesley Longman, 2005.

<sup>28</sup> Franz Baader – Diego Calvanese – Deborah L. McGuinness – Daniele Nardi – Peter F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge: University Press, 2003.

usate le mappe concettuali<sup>29</sup> per la loro semplicità e perché appaiono di immediata comprensione. Le mappe concettuali sono strumenti per organizzare e rappresentare la conoscenza in termini di concetti, o entità, e relazioni fra concetti, o proposizioni. I concetti sono usati per rappresentare le proprietà standard o *commonalities* di una classe di eventi o oggetti. Le proposizioni sono asserzioni su oggetti o eventi che appartengono all'universo che si vuole rappresentare. Nella rappresentazione grafica, i concetti sono iscritti in cerchi o in riquadri, mentre le proposizioni sono rappresentate da linee (freccie) che connettono i concetti da mettere in relazione, ognuna etichettata da parole esprimenti il significato della relazione.

Le entità inizialmente introdotte nel *Manifesto* sono state organizzate in una gerarchia di domini, cioè gruppi di concetti e relazioni, ognuno identificato da un nome, che modellano specifici aspetti dei sistemi di tale mondo. In tal modo ciascuno dei tre "sistemi" in cui si articola il mondo delle biblioteche digitali (nel seguito useremo il termine DLS per indicare tali sistemi in modo indifferenziato) può essere modellato attraverso le entità e le relazioni catturate da questi domini a diversi livelli di astrazione, come mostrato nella Figura 7. Complessivamente il modello consiste di 218 concetti e 52 relazioni che per ovvi motivi di spazio non saranno dettagliatamente descritte in questo articolo.

Il dominio più alto nella gerarchia è Digital Library Domain e concettualmente racchiude tutti gli elementi necessari a rappresentare i tre sistemi per biblioteche digitali. Questo dominio è diviso in due classi principali: Resource Domain e Complementary Domain.

Il Resource Domain, contiene tutti gli elementi fondamentali del mondo delle biblioteche digitali, identificati dai concetti considerati come fondamentali nel paragrafo 0 del *Manifesto* ed è pertanto suddiviso nei domini: Content Domain, User Domain, Functionality Domain, Policy Domain, Quality Domain e Architecture Domain.

Il Complementary Domain contiene tutti gli altri domini che benché non costituiscono il centro di interesse delle biblioteche digitali sono tuttavia necessari per rappresentarne i sistemi. Il concetto di "dominio complementare" serve per collocare domini diversi da quelli riconosciuti come "fondamentali" e per dare spazio a future estensioni del modello. Esso include concetti quali: Time Domain (cioè concetti e relazioni necessari a catturare aspetti della sfera temporale, quali periodi e intervalli di tempo); Space Domain (cioè concetti e relazioni necessari a catturare aspetti della sfera fisica, quali regioni e località); Language Domain (cioè concetti e relazioni necessari a catturare aspetti della comunicazione parlata o scritta, consistente nell'uso di parole in forma strutturata e convenzionale).

<sup>29</sup> Joseph D. Novak – D. Bob Gowin, *Learning How to Learn*, Cambridge: Cambridge University Press, 1984; Joseph D. Novak – Alberto J. Cañas, *The Theory Underlying Concept Maps and How to Construct Them: Technical Report IHMC CmapTools 2006-01*, Pensacola (FL): Florida Institute for Human and Machine Cognition, 2006.



Figura 7. Mappa concettuale dei domini delle biblioteche digitali

## Il Resource Domain

Il Resource Domain rappresenta tutte le entità e le relazioni gestite nel mondo delle biblioteche digitali. Il concetto principale in questo dominio è quello di risorsa (Resource), che include qualsiasi entità compresa in questo mondo (Figura 8). La nozione di “risorsa” quale concetto primitivo in un certo dominio non è nuova. Nel contesto del Web, ad esempio, “risorsa” è la nozione primitiva nell’intera architettura<sup>30</sup>. La nozione di “risorsa Web” si è evoluta dalla sua originaria identificazione con un documento o un file per arrivare all’attuale definizione che comprende qualsiasi entità identificabile, nominabile o indirizzabile nel Web.

Questa nuova concezione si accorda benissimo con il significato assunto dallo stesso termine nel mondo delle biblioteche digitali.

Istanziamenti del concetto di “risorsa” nel mondo delle biblioteche digitali sono quelle di Information Object in qualsiasi forma (per esempio documenti, immagini, video, oggetti informativi compositi, annotazioni, pacchetti di metadati, basi di dati, *streams*, interrogazioni e loro risultati), Actor (sia umani che entità inanimate), Function, Policy, Quality Parameter e Architectural Component. Ciascuno di questi concetti è il concetto principale nei rispettivi domini Content, User, Functionality, Policy, Quality e Architecture, quindi ciascun Domain consiste di

<sup>30</sup> *Architecture of the World Wide Web, Volume One*, edited by Ian Jacobs, Norman Walsh, W3C Recommendation, <http://www.w3.org/TR/webarch>.

Resource, e le Resource sono i mattoni di tutti i Digital Library Domain. Tutti i diversi tipi di Resource condividono molte caratteristiche e modalità di relazione con altre Resource. Ciascuna Resource è:

- identificata da un identificativo unico Resource Identifier (<identifiedBy>);
- predisposta secondo un certo Resource Format (<hasFormat>); tale formato può essere derivato da una ontologia, al fine di assicurarne una interpretazione univoca, oppure può essere costruito in modo arbitrariamente complesso e strutturato poiché una Resource può essere un oggetto composito, costituito cioè da una moltitudine di Resource (<hasPart>) e connessa ad altre Resource (<associatedWith>);
- caratterizzata da vari Quality Parameter, ciascuno relativo al comportamento della risorsa rispetto a un certo attributo (<hasQuality>);
- regolata da Policy (<regulatedBy>) riguardanti ogni aspetto del suo ciclo di vita;
- espressa da (<expressedBy>) un Information Object (ad esempio una risorsa del dominio Policy è espressa da un oggetto informativo che contiene una descrizione testuale della policy oppure un diagramma di flusso che caratterizza la policy);
- descritta o commentata attraverso un Information Object, in particolare attraverso Metadata (<hasMetadata>) e Annotations (<hasAnnotation>).

Per quanto riguarda la loro organizzazione, le Resource possono essere raggruppate in Resource Set (<belongTo>), cioè gruppi di Resource da considerarsi come una singola entità per specifici scopi gestionali o applicativi. Esempi di Resource Set nei vari domini sono Collection, nel Content Domain, o Group nello User Domain.

Il concetto di Resource permette di modellare con un alto livello di astrazione le caratteristiche condivise da tutte le entità del mondo delle biblioteche digitali, e rappresentare tipi di entità più specifici attraverso l'ereditarietà delle caratteristiche condivise; si ottiene così un modello conciso ed elegante che permette una efficiente implementazione e interfacce-utenti uniformi. I vantaggi di questa modalità di modellazione possono tradursi in caratteristiche innovative dei sistemi, sia funzionali che implementative. Per esempio, uno dei risultati può essere un meccanismo unificato per trattare relazioni e funzioni che riguardano tutti i tipi di risorse oppure funzionalità di ricerca per scoprire in modo uniforme le varie entità presenti in una DL.

## Il Content Domain

Il Content Domain rappresenta tutte le entità relative alle informazioni che i DLS gestiscono per soddisfare le esigenze informative dei loro utenti. Il concetto caratterizzante questo dominio è Information Object, che è una Resource. Un



Figura 8. Mappa concettuale del dominio DL Resource<sup>31</sup>

Information Object rappresenta qualsiasi unità di informazione gestita nei DLS, comprendendo documenti testuali, immagini, documenti audio, documenti multimediali e oggetti tridimensionali, come pure basi di dati e archivi di dati. Questo concetto include anche oggetti composti e Collection di Information Object. Gli Information Object possono essere classificati in base alle seguenti caratteristiche:

- Per tipo di dati, informazioni o conoscenze da essi contenuti, e precisamente:
  - oggetti contenenti dati “grezzi”, provenienti direttamente dal mondo esterno (in particolare dati o flussi di dati registrati da strumenti e sensori di vario genere). Per elaborare o interpretare questi dati non strutturati è spesso necessario che essi siano accompagnati da metadati;
  - oggetti contenenti dati elaborati che danno luogo a un risultato spesso chiamato informazione (in contrapposizione a dati “grezzi”) o conoscenza.
- Per tipo di rappresentazione o codifica, e precisamente:
  - oggetti in cui l’informazione/conoscenza è codificata in linguaggio naturale e incorporata in un documento. In senso lato, questa tipologia include anche illustrazioni e documenti audio;
  - oggetti in cui l’informazione/conoscenza è codificata in una struttura formale, ad esempio nelle tabelle di una base di dati o utilizzando gli elementi di una ontologia.
- Per caratteristiche della rappresentazione digitale, e precisamente:
  - oggetti creati originariamente in forma digitale, quali testi nati in forma digitale o immagini catturate da una macchina da ripresa digitale;

<sup>31</sup> Un concetto connesso ad una relazione attraverso una linea tratteggiata rappresenta un attributo della relazione. In questa figura, la linea tratteggiata che qualifica la relazione “associatedWith” significa che l’associazione ha un certo scopo (Purpose).

- oggetti prodotti dalla digitalizzazione di oggetti non digitali;
- oggetti non digitali rappresentati da un record di metadati.

Poiché un Information Object è una Resource, di questa esso eredita tutte le caratteristiche; precisamente:

- ha un identificatore unico (Resource Identifier), chiamato anche identificatore dell'oggetto informativo;
- è predisposto secondo un formato (Resource Format), chiamato anche modello del documento;
- può essere composto in modo arbitrario, così da poter catturare artefatti per natura compositi;
- è caratterizzato da vari Quality Parameter, ognuno relativo a un suo differente aspetto qualitativo;
- è regolato da Policies che determinano ciascun aspetto del suo ciclo di vita;
- può essere descritto o arricchito da Metadata e Annotations.

Gli Information Object possono essere rappresentati in modo più specializzato in corrispondenza al livello di astrazione con cui sono specificati, ricorrendo cioè al concetto di Information Object by Level of Abstraction. La specializzazione di questo concetto può essere ottenuta con uno qualsiasi dei modelli esistenti, ad esempio il modello IFLA FRBR (International Federation of Library Associations and Institutions Functional Requirements for Bibliographic Records)<sup>32</sup> che distingue fra:



Figura 9. Mappa concettuale del dominio Content

<sup>32</sup> IFLA Study Group on the Functional Requirements for Bibliographic Records, *Functional Requirements for Bibliographic Records: Final Report*, Munchen: K. G. Saur, 1998, <http://www.ifla.org/VII/s13/frbr/frbr.htm>.

- Work (opera), cioè una specifica creazione artistica o intellettuale, ad esempio un'opera di Shakespeare;
- Expression (espressione), l'espressione artistica o intellettuale di un'opera, che può avvenire in diverse forme, ad esempio l'espressione di un'opera di Shakespeare può essere una lettura o una rappresentazione teatrale;
- Manifestation (manifestazione), la realizzazione fisica di una particolare espressione di un'opera, ad esempio la manifestazione di una rappresentazione teatrale può essere una registrazione audio o un video;
- Item (copia), la singola copia di una manifestazione.

Gli Information Object possono essere specializzati anche in relazione al ruolo predominante svolto da essi in relazione ad altri oggetti. Così il concetto di Information Object by Relationship è il contenitore delle classi in cui possono essere suddivisi gli oggetti, e precisamente:

- Primary Information Object, cioè la classe degli oggetti che hanno vita propria, come un libro o un archivio di dati;
- Metadata Object, cioè la classe degli Information Object il cui scopo principale è fornire informazioni su una specifica Resource (nella maggior parte dei casi un Primary information object);
- Annotation Object, cioè la classe degli Information Object il cui scopo principale è annotare una specifica Resource (o una sua Region). Esempi di Annotation Object sono note, commenti strutturati e collegamenti. Gli Annotation Object aiutano nella interpretazione della Resource a cui si riferiscono, oppure danno spiegazioni dettagliate o presentano critiche.

Questo modello di classificazione è il riflesso di una intuizione che rende il Reference Model diverso dalla maggior parte dei modelli o standard *de facto* presenti nel mondo delle biblioteche digitali, e precisamente l'intuizione che un Information Object non nasce come Metadata o Annotation (per fare un esempio) ma lo diventa in virtù del suo ruolo in relazione ad altri Information Object. In altre parole, l'annosa questione della natura delle annotazioni, se cioè esse siano "contenuto" o metadati, è semplicemente mal posta.

Dal punto di vista organizzativo, gli Information Object possono essere raggruppati in Collection, cioè gruppi di oggetti considerati come un'unica entità per fini gestionali o applicativi. Poiché le Collection sono Information Object, esse ereditano di questi tutti gli aspetti di modellazione e tutte le funzionalità; ad esempio possono essere annotate. Inoltre le Collection sono una specializzazione del concetto di Resource Set. In effetti, le Collection sono caratterizzate secondo due criteri. Il primo riguarda le modalità di raggruppamento e può essere espresso in diversi modi, ad esempio con l'enumerazione esplicita di tutti gli oggetti apparte-

nenti alla collezione oppure attraverso espressioni logiche esprimenti le caratteristiche delle Resource qualificate a far parte della collezione. Il secondo è l'insieme concreto delle risorse che soddisfano il primo criterio (Resource Set). Queste caratteristiche sono diversamente implementate nei diversi sistemi, dando luogo a scenari che possono essere statici o altamente dinamici. Un'altra specializzazione del concetto di Resource Set solitamente associata al Content Domain, è Result Set. Nelle biblioteche digitali tradizionali il Result Set è l'insieme dei documenti recuperati attraverso una Query. In questo contesto esso rappresenta l'insieme delle risorse risultanti da una Query, senza vincoli sul loro tipo.

### Lo User Domain

Lo User Domain rappresenta tutte le entità esterne interagenti con un DLS, cioè persone o entità inanimate come programmi software o strumenti fisici. In questi ultimi sono compresi, ad esempio, i servizi di sottoscrizione automatica offerti dalle università ai loro studenti per accedere ai contenuti di una biblioteca digitale esterna; fra gli User di un DLS può essere compreso perfino un altro DLS. L'inclusione di hardware e software fra le entità che possono interagire con i DLS è una innovazione significativa rispetto a quanto fatto in altri modelli del mondo delle biblioteche digitali e ne riflette l'ampiezza della conoscenza e della concettualizzazione. Per catturare questo più ampio significato della parola "user", è stato usato il concetto di Actor come elemento principale di questo dominio.

Essendo una Resource, un Actor eredita di questa tutte le caratteristiche principali; esso cioè:

- ha un identificatore univoco (Resource Identifier), chiamato anche identificatore di utente;
- è predisposto secondo un formato (Resource Format), chiamato anche profilo utente;
- può essere organizzato in gruppi, arbitrariamente complessi, per le caratteristiche di composizione e collegamento tipiche di ciascuna Resource. Il significato di questa caratteristica può essere compreso facendo riferimento alla Figura 10. Ad esempio, collaborazioni fra utenti o paternità collettive di un'opera possono essere rappresentate dando valori appropriati all'attributo Purpose della relazione <associatedWith>;
- è caratterizzato da vari Quality Parameter, ognuno relativo ad un suo differente aspetto qualitativo;
- è regolato da Policy che governano ciascun aspetto del suo ciclo di vita; ad esempio, le regole che definiscono quali Function un Actor può esercitare e a quali Information Object può avere accesso;
- può essere arricchito da Metadata e Annotation; ad esempio, un certo esemplare di Actor può attribuire a un diverso esemplare la caratteristica di "amico".



In un DLS un Actor è rappresentato attraverso un Actor Profile ed interagisce con il sistema stesso attraverso un insieme di Functions. L'Actor Profile è un Information Object che rappresenta il concetto di Actor catturandone una grande varietà di caratteristiche. Questo può essere importante per un particolare DLS perché permette di rappresentare, in modo personalizzato, il fatto che un Actor può usare il sistema e interagire sia con il sistema sia con altri Actor in un modo fortemente personalizzato. Così l'Actor Profile non solo serve a rappresentare un Actor all'interno del sistema, ma fissa anche le Policy e i Role che determinano quali Functions gli sono permesse e su quali Resource può contare durante il suo ciclo di vita. Per esempio, un particolare esemplare di Actor può essere autorizzato a fare ricerche all'interno di particolari Collection o a collaborare con altri particolari Actor. Le caratteristiche rappresentate in un Actor Profile variano in dipendenza del fatto che l'Actor sia o no un umano e possono includere: dati anagrafici (età residenza, livello di istruzione, ecc. per gli umani, sistema operativo o edizione del Server Web per il software) e preferenze (ad esempio settori di interesse per ambedue).

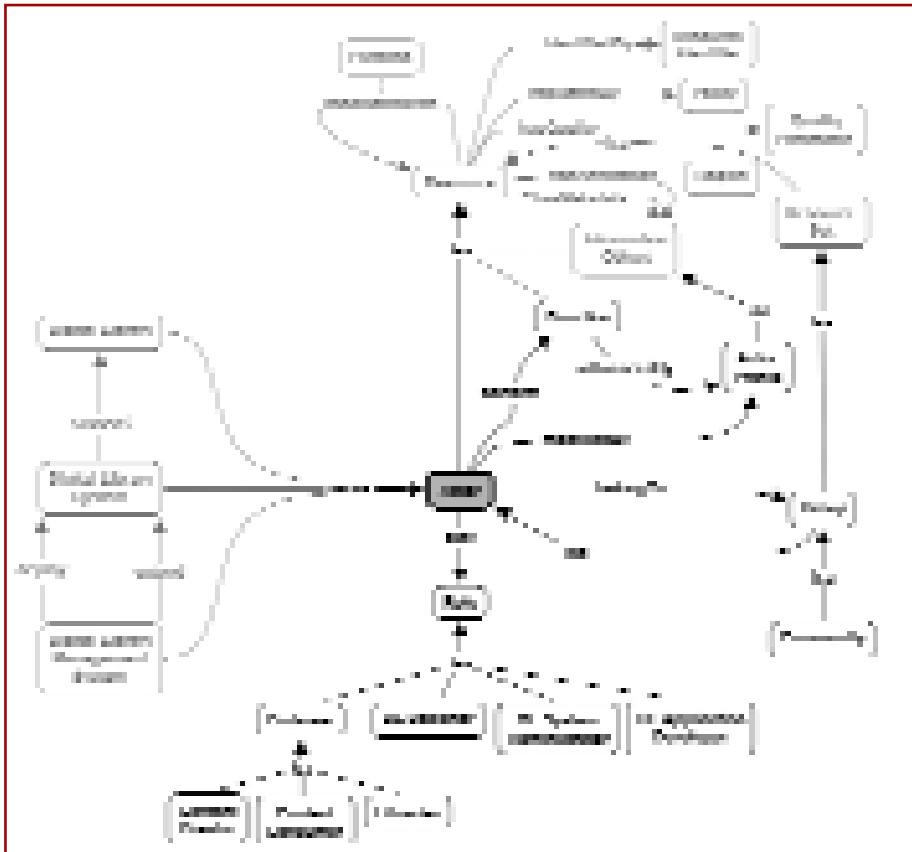


Figura 10. Mappa concettuale del dominio User

Un Actor può comportarsi in modi diversi, cioè avere diversi Role, in tempi diversi, e questa è un'altra deviazione dagli approcci tradizionali, nei quali esistono limiti invalicabili tra i diversi Role e ciascun Actor può avere soltanto uno di essi. Importanti categorie di Actor Role sono indicate nel paragrafo 0: End-user, Designer, System Administrator, e Application Developer. Durante il ciclo di vita dei DLS ciascuno di questi ruoli può essere svolto in modo complementare a un altro.

Un Actor può far parte di un Group. Un Group rappresenta un insieme di Actor che, condividendo molte delle loro caratteristiche, possono essere considerati come un unico Actor con un suo profilo e con un suo identificatore. I membri di un Group ereditano, in tutto o in parte, le caratteristiche del Group a cui appartengono, quali ad esempio interessi, Policy e Role, ma possono avere caratteristiche aggiuntive in conformità con il loro profilo individuale. Una particolare sottoclasse di Group è Community, che denota un gruppo di persone con interessi comuni. In una Community possono esistere intenti, opinioni, risorse, preferenze, bisogni e rischi comuni ed altre comuni condizioni che definiscono l'identità dei partecipanti e il loro grado di coesione.

### **Il Functionality Domain**

Il Functionality Domain rappresenta l'ambito più ricco del mondo delle biblioteche digitali e il più aperto alle innovazioni, poiché cattura tutte le possibili elaborazioni eseguibili sulle Resource e tutte le attività osservabili in un DLS da parte di Actor. Il concetto principale in questo dominio è Function, cioè una particolare elaborazione che può essere eseguita su una Resource o su un Resource Set come risultato dell'attività di un particolare Actor. È importante notare che questa definizione di Function si basa sul concetto generale di Actor, che comprende non solo utenti umani ma anche entità inanimate, e sul concetto di Resource, che comprende tutte le entità coinvolte in un DLS o da esso influenzate ponendo così il concetto di Functionality di un DLS in una nuova prospettiva. Infatti mentre nelle biblioteche digitali di concezione tradizionale le funzioni sono tipicamente associate al contenuto ed eseguite da umani, in questa nuova prospettiva una Function può essere eseguita anche da non-umani e su qualsiasi tipo di Resource. Ad esempio, un utente può non solo fare ricerche sul contenuto di un DLS, ma può anche fare ricerche su altri Actor, così come un programma può andare alla ricerca di Function disponibili, ecc.

Nel Reference Model, ciascuna Function è essa stessa una Resource, e perciò di essa eredita tutte le caratteristiche:

- ha un identificatore univoco (Resource Identifier);
- può essere strutturata in flussi di lavoro arbitrariamente complessi in virtù delle caratteristiche di composizione e collegamento tipiche di ciascuna Resource; ad esempio, può essere strutturata come funzione composta risul-

- tante dalla combinazione di più sotto-funzioni;
- è caratterizzata da vari Quality Parameter, ognuno relativo ad un suo differente aspetto qualitativo;
- il suo ciclo di vita e il suo comportamento sono regolati da Policy; ad esempio, quelle che definiscono quali Actor sono autorizzati a eseguire quella Function in un certo contesto;
- può essere arricchita da Metadata e Annotation;

Nel concetto di Function sono implicitamente comprese elaborazioni di Resource e attività svolte da Actor oltre a funzioni di supporto fornite dal DLS. Per esempio, la funzione di visualizzazione di una lista di risorse presuppone sia che esista una funzione di sistema per generare una lista adatta ad essere visualizzata sia la funzione dell'Actor che scorre la lista.

A parte gli aspetti di modellizzazione, è importante tener presente che l'insieme di Function offerte dai DLS è la diretta conseguenza delle aspettative dei loro Actor. Le Function concorrono a realizzare ciò che viene usualmente chiamato "*business process*", cioè il servizio finalizzato a soddisfare specifici requisiti di "*business*" derivanti dai bisogni di un "cliente". Questi concetti sono mutuati da un lavoro precedente<sup>33</sup> dove essi sono usati per rappresentare le concrete attività di "*business*" di un DLS. Questi tipi di concetti sono particolarmente importanti ma costituiscono un particolare esempio di modellizzazione eseguibile in un *auxiliary domain*, e per questa ragione non sono stati ulteriormente elaborati.

Per l'ampiezza del campo compreso nel concetto di Function, non è possibile enumerare e predire/prevedere tutti i differenti tipi e sfumature delle Function che possono essere incluse nei DLS, ognuno dei quali può avere il proprio insieme di

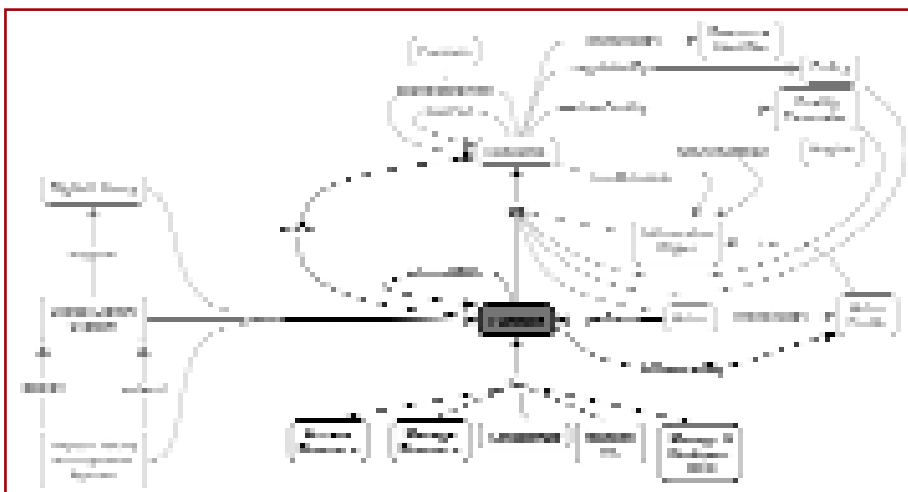


Figura 11. Mappa concettuale del dominio Functionality

Function corrispondentemente ai suoi obiettivi e agli Actor a cui intende fare riferimento. Perciò, Function viene qui specializzata in cinque concetti che, tuttavia, rappresentano ancora classi di attività alquanto generali, come delineato nel seguito.

Il concetto Access Resource comprende tutte le attività legate a richiesta, localizzazione, recupero, trasformazione e, infine, conservazione di Resource.

La caratteristica peculiare del concetto di Access Resource è il fatto di rappresentare Function che non modificano alcun DLS ma servono a identificare Resource da essere soltanto esaminate e percepite da un Actor o, eventualmente, essere ulteriormente usate e gestite attraverso altre Function, chiamate funzioni di Manage Resource. Perciò, fra le funzioni di Access Resource quella centrale è Discover, che agisce su Resource Set per recuperare le Resource desiderate.

Il concetto di Manage Resource comprende tutte le attività relative alla creazione di nuove Resource e al loro inserimento in una DL, alla loro cancellazione, al loro aggiornamento come pure alla loro trasformazione. Un'operazione di trasformazione può dar luogo a nuove Resource che possono essere inserite nel DLS o semplicemente può essere eseguita al momento dell'accesso. Tutte queste attività possono essere rappresentate con Function individuali, specializzate per ciascun tipo di risorsa.

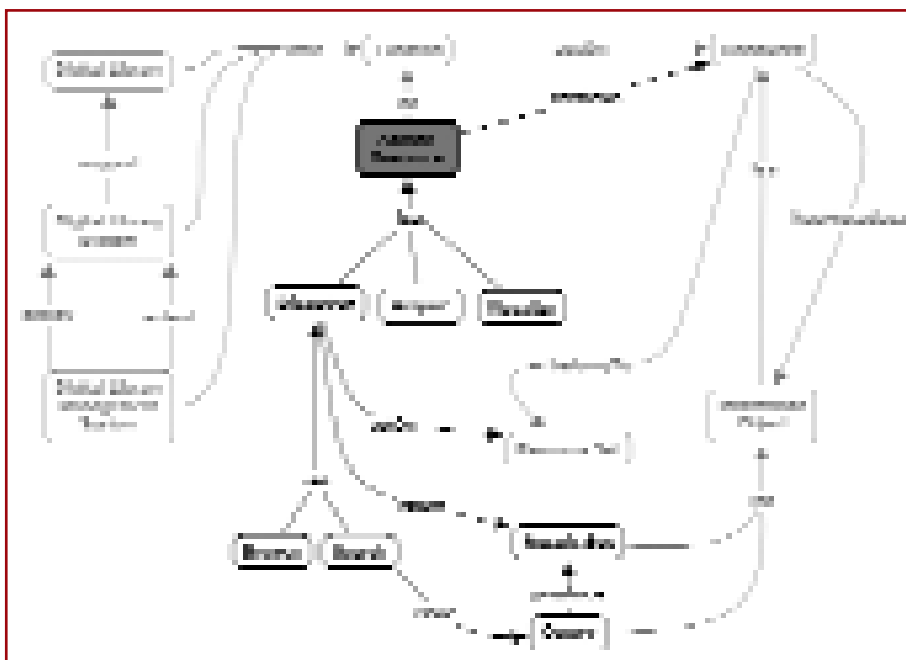


Figura 12. Mappa concettuale delle Access Resource Functions

<sup>33</sup> Brian Lavoie – Geneva Henry – Lorcan Dempsey, *A Service Framework for Libraries*, «D-Lib Magazine», vol. 12, n. 7/8, July/August 2006, <http://www.dlib.org/dlib/july06/lavoie/07lavoie.html>.

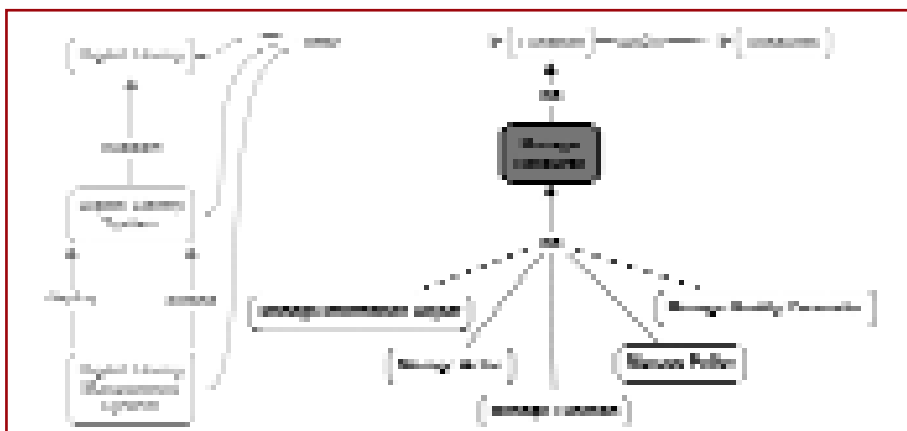


Figura 13. Mappa concettuale delle Manage Resource Functions

Alcune Function possono essere applicate alle Resource, mentre altre possono essere applicate ai metadata che descrivono le Resource. Applicabili a tutte le Resource sono invece le Function per la loro creazione, inserimento, cancellazione, aggiornamento, validazione e annotazione; esse possono essere specializzate per particolari tipi di risorse, come esemplificato nel seguito.

Manage Information Object comprende sia concetti di Function relativi alla creazione, elaborazione e trasformazione di Information Object primari, cioè indipendenti da altri oggetti, ad esempio Author, sia altri concetti che sono invece relativi alla creazione, elaborazione e trasformazione di Information Object che rappresentano altri Information Object o, in generale, altre Resource (quali riferimenti ad altre Resource, composizioni di altre Resource, ecc.).

Manage Actor comprende Function necessarie alla gestione di Actor della DL, quali la loro registrazione, il loro “login” e la personalizzazione delle Function che essi hanno il diritto di usare.

Un'altra specializzazione di Function è strettamente correlata all'User Domain. Questa è la funzione Collaborate, che rappresenta tutte le attività che permettono a più Actor di collaborare attraverso una DL per il raggiungimento di un obiettivo comune.

Le altre specializzazioni del concetto di Function abbracciano tutte le attività legate alla gestione di una biblioteca digitale (DL) intesa come sistema.

Manage DL comprende una grande varietà di Function che offrono strumenti per la gestione quotidiana di una DL rispetto a tutti i domini. Questa classe include sia le funzioni di gestione delle Collection, dei gruppi di utenti e le procedure di affiliazione (membership), sia la gestione generale dei domini “policy”, “quality” e “functionality”.

Manage & Configure DLS comprende Function utili al ruolo di DLS Administrator nella sua attività per costruire, configurare e monitorare la DL negli aspetti fisici, cioè nella scelta di particolari Architectural Components offerti dal DLMS per portare la DL (in realtà il DLS) a uno stato operativo.

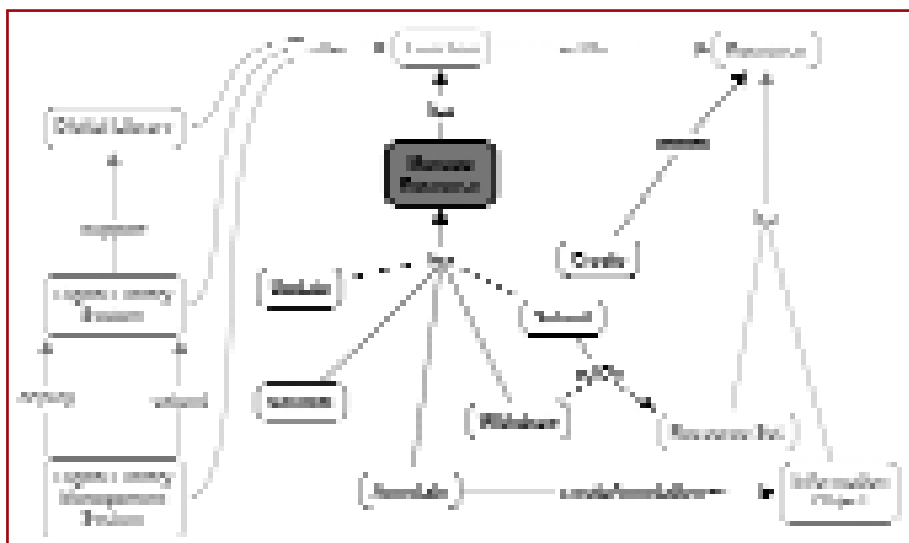


Figura 14. Mappa concettuale di Manage Resource Functions

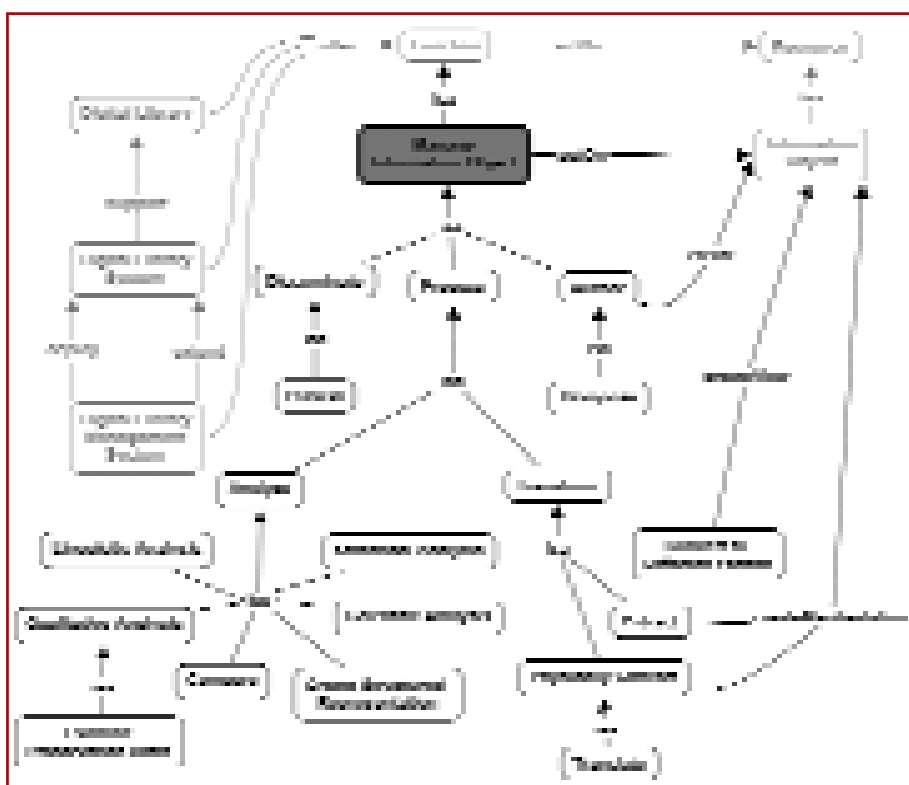


Figura 15. Mappa concettuale di Manage Information Object Functions

Come già osservato, il Functionality Domain è probabilmente il più dinamico di tutti i DL domain; di conseguenza, ciò che è descritto nell'attuale versione del Reference Model rappresenta solo un sottoinsieme delle Function immaginabili per i DLS e corrisponde ai concetti considerati più importanti/decisivi.

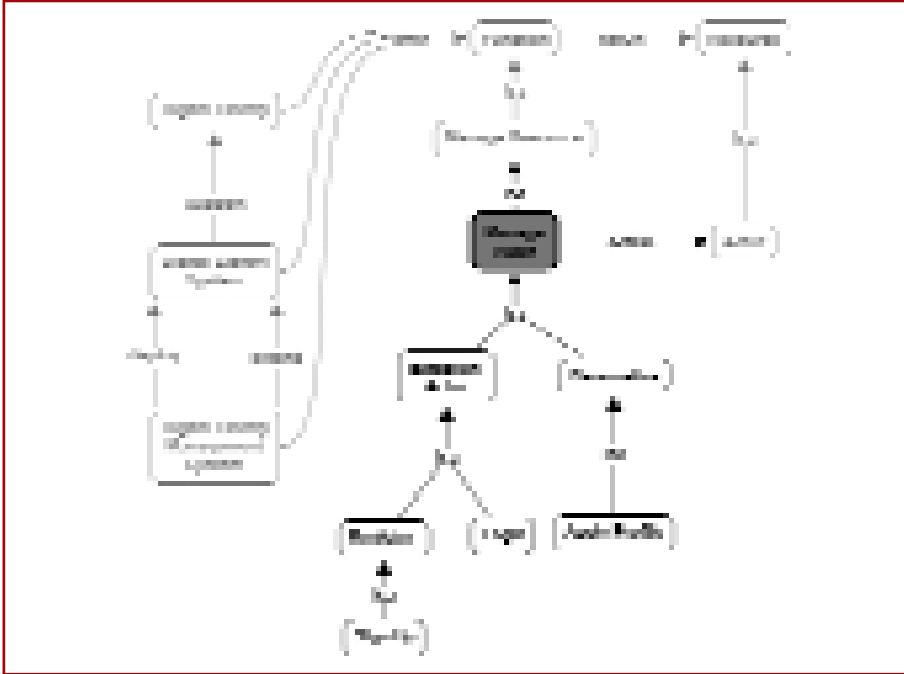


Figura 16. Mappa concettuale di Manage Actor Functions

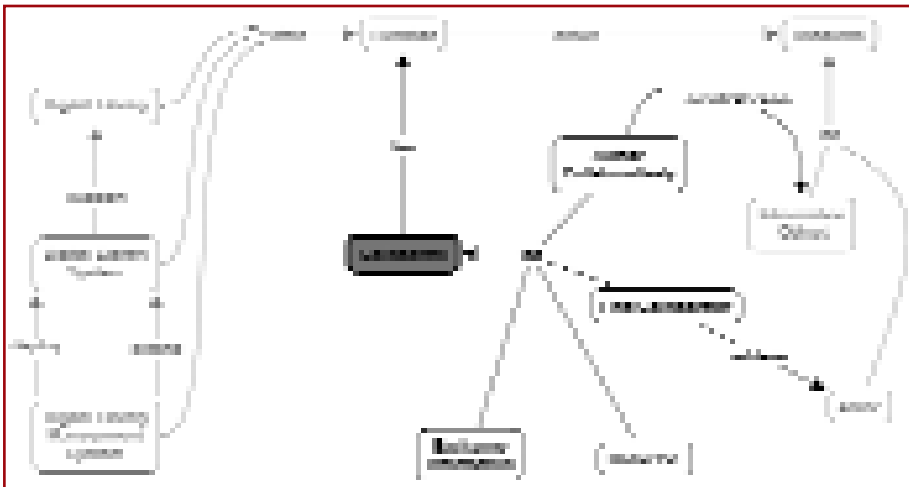


Figura 17. Mappa concettuale di Collaborate Functions

## Il Policy Domain

Il Policy Domain rappresenta l'insieme delle condizioni, norme ed accordi che regolano le operazioni dei DLS. Per sua natura, questo dominio è molto ampio e dinamico, ma la rappresentazione che se ne fornisce non intende essere esaustiva specialmente nei confronti della miriade di regole specifiche che ciascuna istituzione potrebbe voler modellare e applicare. Quindi il Policy Domain cattura soltanto quelle relazioni che lo collegano agli altri domini e presenta il sottoinsieme di regole considerate più importanti nel mondo delle biblioteche digitali. Tuttavia il modello è estensibile e può facilmente accogliere ulteriori concetti che fossero eventualmente necessari.

Il concetto principale è Policy, l'entità che governa una Resource dal punto di vista amministrativo. Nel nostro modello ciascuna Policy è essa stessa una Resource, e pertanto ne eredita tutte le caratteristiche, ossia:

- ha un identificatore univoco (Resource Identifier);
- può essere strutturata in modo arbitrariamente complesso in virtù delle caratteristiche di composizione e collegamento tipiche di ciascuna Resource; ad esempio, si può ottenere una Policy composta combinando opportunamente le Policy che la devono costituire;
- è caratterizzata da vari Quality Parameter, ognuno relativo a un suo differente aspetto qualitativo;
- Può essere regolata da altre Policy, ad esempio per definire quali Actor sono soggetti a certe policy in un certo contesto;

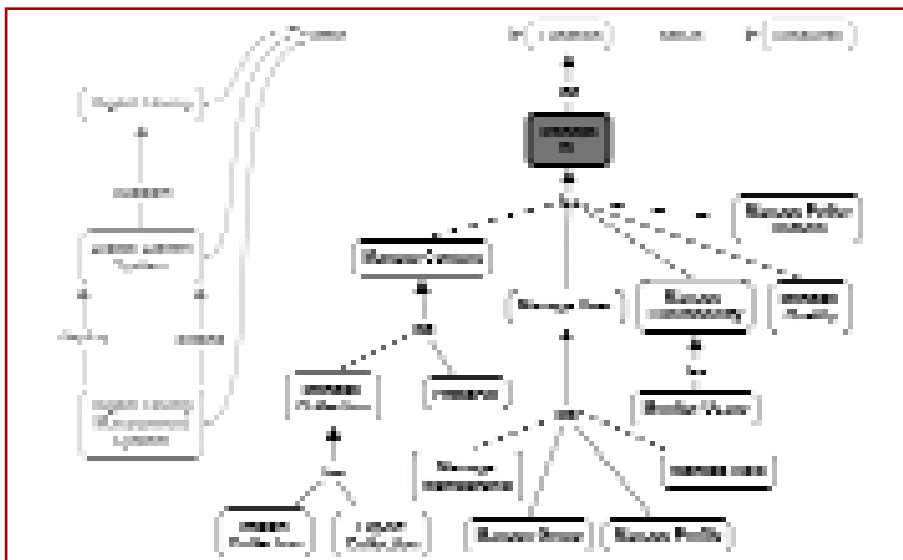


Figura 18. Mappa concettuale di Manage DL Functions



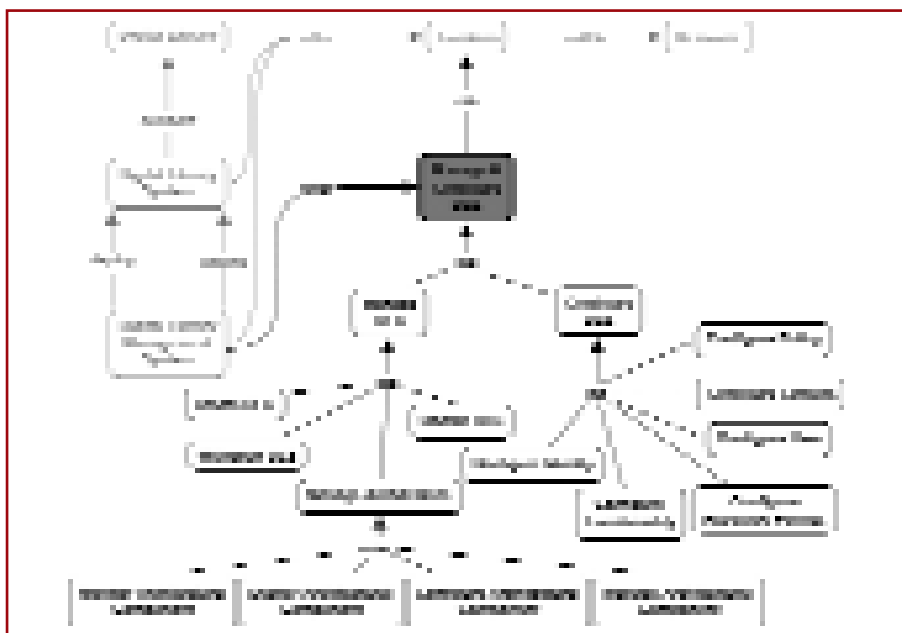


Figura 19. Mappa concettuale di Manage & Configure DLS Functions

- può essere arricchita da Metadata e Annotation.

Policy è in realtà una classe di vari tipi di regolamenti – quelli che vengono più frequentemente applicati nella pratica biblioteconomica. Per i fini della nostra modellizzazione abbiamo identificato due classi “contenitori” ortogonali di concetti, cioè Policy by characteristic e Policy by scope.

Policy by characteristic è ulteriormente specializzata in otto sottoclassi raggruppate in quattro coppie ciascuna delle quali presenta la possibile bipolarità di una Policy: Extrinsic Policy-Intrinsic Policy; Implicit Policy-Explicit Policy; Prescriptive Policy-Descriptive Policy; Enforced Policy-Voluntary Policy. La corretta rappresentazione delle caratteristiche di una specifica Policy favorisce la comprensione del suo significato da parte degli Actor e dei suoi requisiti ai fini della sua traduzione in funzionalità e della sua implementazione all’interno dei diversi sistemi.

Policy by scope è a sua volta specializzata in varie classi, ciascuna relativa a una particolare Policy rispetto a:

- il sistema nella sua interezza, ad esempio Resource Management Policy,
- un certo dominio, ad esempio, User Policy or Content Policy; in alcuni casi, in realtà, una Policy risponde ai bisogni di due domini; ad esempio, Access Policy è allo stesso tempo una User Policy e una Functionality Policy;
- una specifica attività o entità, ad esempio Collection Development Policy.

È importante ricordare, che il modello è estensibile e non intende dare una lista esaustiva delle Policy che regolano il mondo delle biblioteche digitali ma solo rappresentarne alcune fra le più importanti. Fra queste, un posto speciale è occupato da Digital Rights Management Policy and Digital Rights. Nel contesto del nostro modello, la prima governa le Function, mentre la seconda governa gli Information Object.

Dal punto di vista della DL, da quello del DLS, o da quello del DLMS non c'è alcuna differenza di percezione del concetto di Policy ma sono differenti le Resource su cui questi sistemi applicano le Policy. Inoltre la stessa Policy ha diverse materializzazioni nei diversi sistemi, ad esempio un Information Object in una DL è gestito da un servizio del DLS istruito a rilasciare quell'oggetto solo all'Actor che ne è proprietario.

## Il Quality Domain

Il Quality Domain rappresenta quegli aspetti che permettono di considerare i DLS dal punto di vista della qualità, con l'obiettivo di giudicare e valutare tali sistemi rispetto a specifiche caratteristiche. Qualsiasi sistema offre ai suoi Actor un certo livello di Quality. Questo livello di qualità può essere implicitamente concordato, nel senso che gli Actor conoscono quali Quality Parameter il sistema garantisce, oppure è esplicitamente contrattato attraverso un accordo di Qualità del Servizio (QoS).

Il concetto principale di questo dominio è Quality Parameter, cioè l'entità con cui sono rappresentati i diversi aspetti del Quality Domain e sono date informazioni sulle modalità e sulle qualità del comportamento di una Resource sotto un certo aspetto. Infatti, Quality Parameter, insieme ai concetti di Actor, Resource, Measure e Measurement, costituisce la struttura concettuale entro cui trattare le numerose questioni correlate al concetto di qualità. I Quality Parameter esprimono la valutazione che un Actor, umano o non umano, dà della Resource sotto considerazione. I Quality Parameter possono essere valutati secondo diverse Measure, che offrono procedure alternative per valutare i



Figura 20. Mappa concettuale del dominio Policy

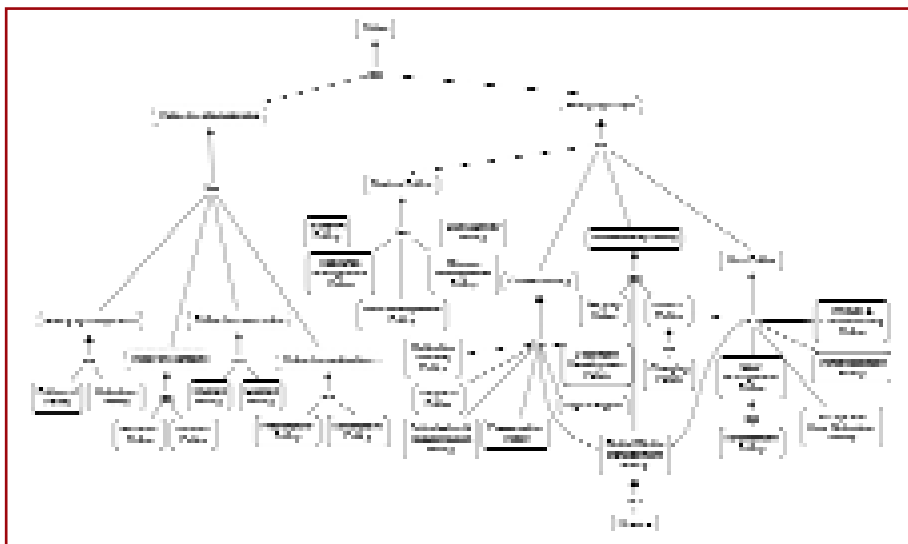


Figura 21. Mappa concettuale della gerarchia di Policy

diversi aspetti di ciascun Quality Parameter e dare ad esso un valore. In realtà i Quality Parameter sono valutati da un Measurement, che rappresenta il valore assegnato a un Quality Parameter rispetto ad una determinata Measure.

Ciascun Quality Parameter è esso stesso una Resource, e ne eredita le caratteristiche:

- ha un identificatore univoco (Resource Identifier);
- può essere strutturato in modo arbitrariamente complesso in virtù delle caratteristiche di composizione e collegamento tipiche di ciascuna Resource; ad esempio, un Quality Parameter può essere composto da Quality Parameter più specifici che di esso catturano i particolari aspetti;
- è esso stesso caratterizzato da vari Quality Parameter, cioè è possibile valutare il grado di affidabilità del valore attribuito a un certo Quality Parameter;
- può essere regolato/specificato da Policies;
- può essere arricchito da Metadata e Annotation.

Il Quality Domain è, per sua natura, molto ampio e dinamico. La rappresentazione che ne dà questo modello è perciò estensibile alla miriade di aspetti qualitativi specifici che ciascuna Istituzione volesse modellare. Quality Parameter è infatti una classe di diversi tipi di caratteristiche qualitative, ad esempio quelle che attualmente rappresentano la pratica comune. Questi parametri sono raggruppati secondo la tipologia della Resource sotto osservazione:

- Generic Quality Parameter si applicano a tutte le Resources, o almeno alla maggior parte di esse;

- System Quality Parameter si applicano alla DL, o al DLS o al DLMS;
- Content Quality Parameter si applicano alle Resource del Content Domain, essenzialmente Information Object;
- Functionality Quality Parameter si applicano alle Resource del Functionality Domain, essenzialmente Function;
- User Quality Parameter si applicano alle Resources dello User Domain, essenzialmente Actor;
- Policy Quality Parameter si applicano alle Resource del Policy Domain, essenzialmente Policy;
- Architecture Quality Parameter si applicano agli Architectural Component, cioè Resource appartenenti all'Architecture Domain.

È importante ribadire che questo criterio di raggruppamento prende in considerazione la tipologia della Resource sotto esame, cioè l'oggetto che si sta valutando. In questa prospettiva, l'Actor, inteso come il soggetto che esprime il giudizio, è sempre preso in considerazione e modellato esplicitamente poiché parte integrante della definizione di Quality Parameter. Per questa ragione, ad esempio, il Quality Parameter chiamato User Satisfaction è raggruppato sotto Functionality Quality Parameter perché questo parametro rappresenta il grado di soddisfazione che un Actor (cioè il soggetto che fa la valutazione) esprime quando usa una certa Function (cioè l'oggetto della valutazione). D'altra parte, nel caso del Quality Parameter chiamato User Behaviour l'oggetto della valutazione è un Actor insieme al suo modo di agire rispetto alla Policy chiamata User Behaviour Policy; per questa ragione tale parametro è stato inserito nel gruppo User Quality Parameter.

Non c'è alcuna fondamentale differenza nella percezione del concetto di Quality Parameter dal punto di vista della DL, da quello del DLS, o da quello del DLMS. Tuttavia, ciascuno di questi "sistemi" applica la nozione di Quality Parameter in una prospettiva diversa; ad esempio, Architecture Quality Parameter sono peculiari del DLS e del DLMS. Un'altra differenza consiste nel modo in cui il medesimo Quality Parameter è soddisfatto nei diversi DLS. Ad esempio, se la DL specifica un certo Quality Parameter, è compito del sottostante DLS soddisfare questa richiesta, mentre è responsabilità del DLMS mettere a disposizione le risorse necessarie a soddisfare le aspettative dell'utente, ad esempio implementando la soluzione architeturale appropriata.

### **L'Architecture Domain**

L'Architecture Domain include concetti e relazioni caratteristici dei due sistemi software che hanno un ruolo attivo nel mondo delle biblioteche digitali, e cioè i DLS e i DLMS. Sfortunatamente nel passato l'importanza di questo fondamentale concetto è stata ampiamente sottostimata, mentre invece una chiara comprensione dei sistemi software che implementano le biblioteche digitali fornisce indicazioni e strumenti affidabili per la loro concreta realizzazione. In particolare questo

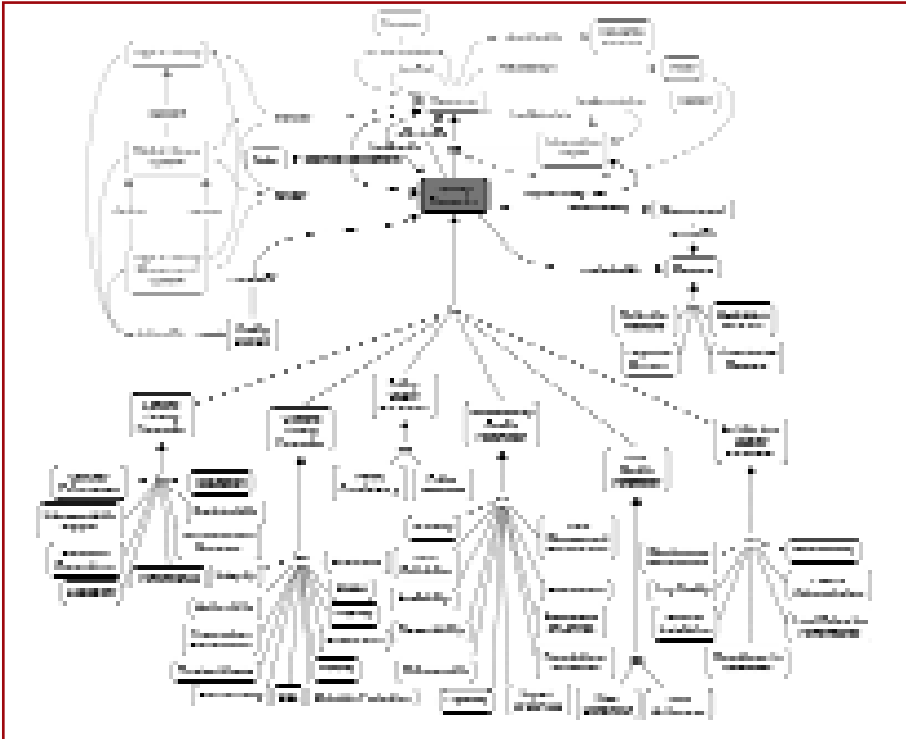


Figura 22. Mappa concettuale del dominio Quality

dominio permette di capire intuitivamente:

- come sviluppare nuovi sistemi in modo appropriato, massimizzando cioè la condivisione e il riuso di risorse per minimizzare costi e tempi di realizzazione;
- come migliorare le prestazioni di sistemi già funzionanti promuovendo l'adozione di standard riconosciuti ed accettati, adatti a semplificare problemi di interoperabilità.

Il concetto di architettura di un sistema software è generalmente ben compreso dalla maggioranza degli ingegneri e degli amministratori e sviluppatori di sistemi, ma non è facilmente definibile. Nella loro *An Introduction to Software Architecture*, Garlan and Shaw pongono l'accento su tematiche di progetto suggerendo che l'architettura del software ha come oggetto la struttura del sistema:

«Beyond the algorithms and data structures of the computation, designing and specifying the overall system structure emerges as a new kind of problem. Structural issues include gross organization and global control structure; protocols for communication, synchronization, and data access; assignment of functionality to design elements; physical distribution; composition of design elements; scaling and performance; and selection among design alternatives»<sup>34</sup>.

Tuttavia l'IEEE (Institute of Electrical and Electronics Engineers, Inc.) Working Group on Architecture<sup>35</sup> dichiara che l'architettura comprende ben più che struttura, e che essa può essere definita come «the highest-level concept of a system in its environment». Questo gruppo, cioè, non si limita a considerare l'architettura di un sistema software come il “centro” di un sistema, ma propone che essa includa l'intero sistema, comprese le modalità del suo uso e il suo ambiente di sviluppo. Per gli scopi del Reference Model, l'architettura di un sistema software è definita come l'organizzazione o struttura dei componenti significativi del sistema (Architectural Component) che interagiscono l'uno con l'altro attraverso le loro interfacce. Questi componenti possono essere composti, a loro volta, di componenti sempre più piccoli insieme alle loro interfacce; tuttavia Architectural Component diversi possono essere fra loro incompatibili, non possono, cioè, coesistere nel contesto dello stesso sistema. Quando nell'industria del software e nella letteratura si usa il termine “componente” ci si riferisce a molti concetti, fra loro diversi. Qui noi usiamo il termine “componente” per indicare una parte del sistema – idealmente una parte non triviale del sistema – quasi indipendente e perciò sostituibile, che svolge una chiara funzione nel contesto di una ben definita architettura.

Ciascun Architectural Component è una Resource, e pertanto ne eredita gli aspetti caratteristici; ad esempio l'essere univocamente identificata. Come qualsiasi Resource, i componenti hanno Metadata (Component Profile) che danno informazioni necessarie alle loro gestione quali: Function implementate o supportate, Interface implementate, Policy da cui sono governati, e Quality Parameter che, per i vari aspetti qualitativi, descrivono le modalità e la qualità del comportamento del componente rispetto a un certo punto di vista.

Gli Architectural Component interagiscono attraverso una Framework Specification alla quale devono essere conformi. Questa Framework Specification definisce quali Interface devono essere implementate dai componenti e i protocolli che governano le interazioni fra i componenti stessi.

Gli Architectural Component sono classificati in Software Architecture Component e System Architecture Component. Queste classi sono usate per descrivere, rispettivamente, la Software Architecture e la System Architecture di un sistema software. I Software Architecture Component sono realizzati da Software Component. Per ciascun Software Component valgono le seguenti affermazioni:

- il Software Component costituisce l'implementazione di una parte del sistema software (che cattura aspetti dei domini Content, User, Functionality, Policy o Quality);

<sup>34</sup> David Garlan – Mary Shaw, *An Introduction to Software Architecture*, CMU/SEI-94-TR-21, [http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/intro\\_softarch.pdf](http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/intro_softarch.pdf).

<sup>35</sup> *IEEE Recommended Practice for Architectural Description*, IEEE Standard Description 1471-2000 (2000).

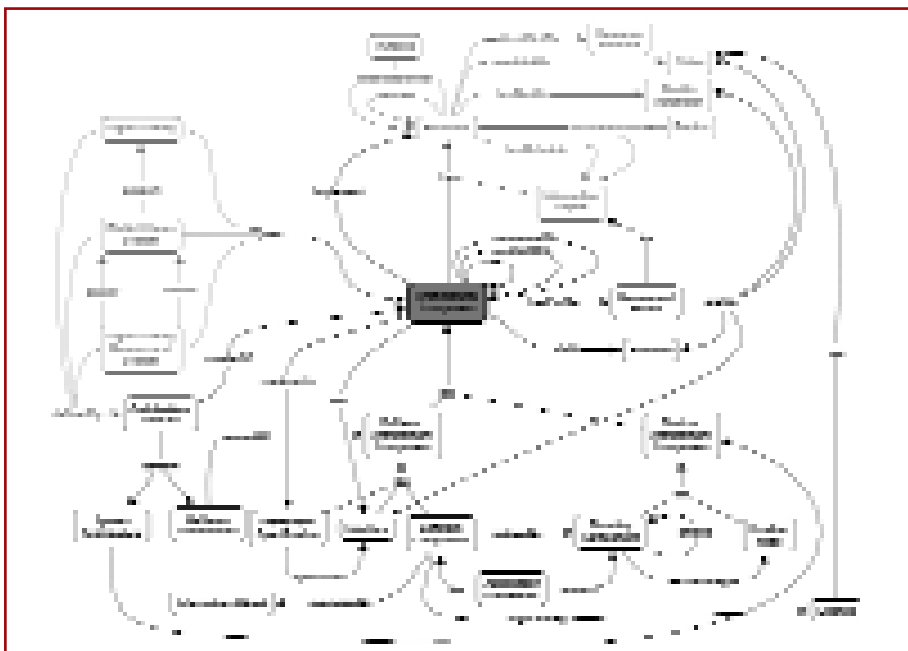


Figura 23. Mappa concettuale del dominio Architecture

- il suo uso è regolato da particolari Policy (License);
- è rappresentato da un Information Object.

I System Architecture Component sono realizzati da Hosting Node e Running Component. Un Hosting Node costituisce l'implementazione dell'ambiente necessario ad ospitare e mantenere funzionanti Software Component. Un Running Component è un esemplare di Software Component attivo in un Hosting Node.

L'insieme dei Software Architectural Component – tra loro interagenti – e l'insieme dei System Architecture Component – tra loro interagenti – rappresentano, rispettivamente, l'aspetto statico e l'aspetto dinamico dei sistemi DLS e DLMS.

Benché la System Architecture di un DLS e la System Architecture di un DLMS siano rappresentabili con lo stesso insieme di concetti e relazioni, i due sistemi sono estremamente diversi e giocano diversi ruoli nel mondo delle biblioteche digitali. Gli aspetti che distinguono un DLS da un DLMS, dal punto di vista dell'architettura, risiedono nel concreto insieme di Architectural Component (in particolare Software Component) che costituisce tali sistemi.

## Conclusioni

Dopo aver illustrato il *Manifesto* delle Biblioteche Digitali, abbiamo presentato il DELOS Digital Library Reference Model nei principi secondo i quali sono stati identificati e organizzati i suoi elementi costitutivi. Abbiamo inoltre descritto i concetti-base e le loro relazioni che catturano l'intrinseca natura del mondo delle biblioteche digitali. Malgrado la complessità di alcuni dei vari aspetti illustrati, si può rilevare come nella maggior parte dei casi sono bastati alcuni concetti e relazioni di base per catturare, di questo mondo, le caratteristiche essenziali.

La struttura concettuale che costituisce il modello può essere usata come strumento per coordinare concezioni, soluzioni e sistemi relativi al mondo delle biblioteche digitali.

Per sua stessa natura il Reference Model è destinato ad evolvere nel tempo sino a raggiungere una forma stabile. In particolare, nell'ambito del progetto DL.org<sup>36</sup> il Reference Model sarà consolidato ed ampliato come risultato di un suo uso estensivo nelle varie attività che il progetto stesso ha posto in essere per poter disegnare soluzioni innovative ai problemi di interoperabilità tra sistemi per biblioteche digitali.

*Digital Libraries are a complex universe. The evolution of this universe, in terms of approaches, solutions and systems, has made it necessary to develop a foundational theory capable of unifying and organizing the overall body of knowledge accumulated in the sector. The DELOS Digital Library Reference Model represents a first contribution towards the development of such a foundational theory, and is the result of the 'collective understanding' matured by the community of scholars who have been contributing to the growth of the sector. The DELOS Digital Library Reference Model identifies a set of notions and relations that are typical of the Digital Libraries arena. This model represents a reference guide for all professionals interested in the development and use of this kind of systems.*

*The present article introduces a Manifesto for Digital Libraries – i.e. a document which publicly defines the aim and strategy for developing of a foundational theory – and illustrates the DELOS Reference Model – i.e. a document which represents the first step in the development of a foundational theory, by introducing the main notions and relations which define the various entities of a Digital Library.*

<sup>36</sup> Digital Library Interoperability, Best Practices and Modelling Foundations, <http://www.dlorg.eu>.