ECODIGIT

Ecosistema Digitale per la Fruizione e la Valorizzazione
dei Beni e delle Attività Culturali della Regione Lazio

D3.3 Studio sugli strumenti di supporto

Acronimo Progetto:	**EcoDigit**
Titolo Progetto:	**Ecosistema digitale per la fruizione e la valorizzazione dei beni e delle attività culturali della regione Lazio**

D3.3

Work Package:	WP3	
Data di Sottomissione:		2 Ottobre 2019
Inizio Progetto:		2 Ottobre 2018
Durata Progetto:		15 Mesi
Reponsabile Deliverable:		Massimo Mecella mecella@diag.uniroma1.it
Versione:		v1.0
Stato:		Versione Finale
Autore:	Luigi Asprino	ISTC-CNR
	Ludovica Marinucci	ISTC-CNR
	Andrea Giovanni Nuzzolese	ISTC-CNR
	Valentina Presutti	ISTC-CNR
Altri contribuenti al lavoro riportato nel deliverable:	Massimo Mecella	RM1
	Marco Canciani	RM3
	Federico Meschini	UNITUS
Reviewer:	Massimo Mecella	RM1
	Miguel Ceriani	RM1

Per citare questo documento si prega di utilizzare il seguente record bibliografico

Luigi Asprino, Ludovica Marinucci, Andrea Giovanni Nuzzolese e Valentina Presutti. *D3.3 Studio sugli strumenti di supporto*. Deliverable Progetto EcoDigit. 2019

Revisioni			
Versione	**Data**	**Modificata da**	**Commento**
0.1	12/9/2019	Luigi Asprino	Creazione Documento
0.2	16/9/2019	Luigi Asprino	Prima Bozza
0.3	20/9/2019	Ludovica Marinucci, Andrea Giovanni Nuzzolese e Valentina Presutti	Prima Revisione
0.4	26/9/2019	Massimo Mecella, Miguel Ceriani	Seconda Revisione

Executive Summary

Le cinque università statali del Lazio in rete con CNR, ENEA e INFN si candidano a costituire il Centro di Eccellenza del Distretto Tecnologico per i beni e le attività Culturali (DTC) del Lazio. La mission del Centro è costituire un centro di aggregazione ed integrazione di competenze nel settore delle tecnologie per i beni e le attività culturali. In questo contesto, il progetto **EcoDigit-Ecosistema digitale per la fruizione e la valorizzazione dei beni e delle attività culturali del Lazio** ha l'obiettivo di arricchire il sistema Anagrafe delle Competenze del DTC con una piattaforma middleware che faciliti l'integrazione di nuove sorgenti di dati e consenta la pubblicazione e il riuso di servizi per la valorizzazione e la fruizione del patrimonio culturale del Lazio.

Il Work Package 3 del progetto si occupa: *(i)* di identificare le potenziali sorgenti (T3.1); *(ii)* di definire un modello di ingresso che le sorgenti dovranno rispettare per poter essere integrate e di selezionare degli strumenti metodologici e tecnologici che una sorgente può adottare per rendere i propri dati compatibili con il modello di ingresso (T3.2); *(iii)* di dimostrare la fattibilità dell'approccio di integrazione proposto (T3.3).

Il presente deliverable, frutto di parte del lavoro del task T3.2, ha l'obiettivo di identificare potenziali strumenti concettuali, metodologici e tecnologici che una sorgente può adottare per rendere i propri dati compatibili al modello di ingresso atteso dalla Digital Library e descritto nel Deliverable D3.2 [1]. Il lavoro descritto in questo deliverable si è avvalso dei risultati dei task: "T2.2 Architettura e specifica della piattaforma middleware" e "T3.1 Censimento ed individuazione delle sorgenti potenziali".

Indice

Elenco delle figure

Elenco delle tabelle

1 Introduzione

1.1 Obiettivi Work Package

Il **WP3 "Modelli, metodi e strumenti per l'aggregazione di sorgenti"** si occupa dei contenuti di EcoDigit. In esso sono curati: le sorgenti dei dati, i modelli e le tecniche per la loro integrazione e standardizzazione, basata su formati aperti e semantici.

Obiettivo di questo WP è analizzare, progettare e sviluppare metodologie e strumenti per aggregare risorse e poli distinti sul territorio (indicate genericamente come sorgenti), che includano archivi, dati strutturati e database, patrimoni fotografici e multimediali in generale, biblioteche digitali, ecc.

1.2 Obiettivo del deliverable

Questo deliverable ha l'obiettivo di identificare potenziali strumenti concettuali, metodologici e tecnici che una sorgente può adottare per rendere i propri dati conformi al modello di ingresso atteso dalla Digital Library e descritto nel Deliverable D3.2 [1].

1.3 Relazione con le altre attività del progetto

Influenza degli altri task del progetto sul lavoro descritto in questo documento. Il lavoro descritto in questo deliverable si è avvalso dei risultati dei task: "T2.2 Architettura e specifica della piattaforma middleware", in quanto gli strumenti identificati in questo documento devono risultare compatibili con l'architettura definita; "T3.1 Censimento ed individuazione delle sorgenti potenziali", in quanto l'identificazione degli strumenti dipende dalla tipologia delle sorgenti identificate (in particolare, dai formati dati supportate dalle sorgenti); "T3.2 Definizione del modello di integrazione di una sorgente", in quanto gli strumenti identificati devono essere in grado di trasformare i dati di una potenziale sorgente nel formato sintattico e semantico atteso in ingresso.

Influenza del lavoro descritto in questo documento sugli altri task del progetto. Gli strumenti identificati in questo documento influenzeranno la progettazione dell'architettura (cf. Task 2.2). Inoltre, il lavoro descritto in questo documento suggerirà una serie di metodologie e tecnologie che potranno essere usate nella proof-of-concept nel task 3.3. Infine, il lavoro descritto in questo documento identifica degli strumenti e delle metodologie per far sì che i dati di alcune sorgenti selezionate possano essere disponibili e integrati nel sistema e quindi fornire dei contenuti utili per la costruzione del prototipo di servizio avanzato per la fruizione dei beni culturali nel dominio della formazione (cf. Task 4.4).

1.4 Outline documento

Nella sezione 2 illustriamo lo scenario di riferimento. La sezione 3 discute le possibili metodologie di integrazione dati che una sorgente può adottare per far rendere i propri dati conformi al modello di ingresso. Nella sezione 4 elenca gli strumenti tecnologici che una sorgente può implementare per trasformare il formato sintattico e semantico dei propri dati. Infine, la sezione 5 sintetizza i risultati e conclude il documento.

2 Scenario di Riferimento

L'obiettivo del middleware EcoDigit è fornire strumenti tecnici per fare in modo che una sorgente dati possa essere integrata nella Digital Library del Centro di Eccellenza. In Figura 1 viene mostrato lo scenario di riferimento del problema di integrazione dati. L'interfaccia di caricamento dati della Digital Library è il componente di Content Management implementato da Apache Manifold[1] o il gestore di code implementato da Apache Active MQ[2]. Entrambi i componenti accettano in input file in formato XML, ognuno dei quali contiene dati e/o metadati relativi ad una risorsa che si vuole inserire nella Digital Library. Il sistema permette anche un caricamento in blocco di più risorse in un singolo file, ma la frammentazione in un file per ogni risorsa permette una più agevole gestione degli aggiornamenti (evita di dover riprocessare tutto il file anche se solo una risorsa è stata modificata).

E' importante sottolineare che con risorsa si intende i dati/metadati relativi ad una qualsiasi entità che sia di una tipologia di interesse per la Digital Library (e.g. Oggetto Fisico nel dominio dei Beni Culturali, una Persona o una Organizzazione che si vuole inserire nell'anagrafe delle competenze del distretto ecc.). I file devono essere strutturati secondo il modello di ingresso definito nel Deliverable D3.2 [1]. Viene considerata sorgente un qualsiasi sistema che esponga dati in maniera strutturata, ad esempio in formato tabellare (e.g. CSV, DBMS) o secondo una struttura a grafo (e.g. RDF). L'obiettivo del middleware Ecodigit è quello di fornire un componente che offra alle sorgenti degli strumenti che permettano di rendere conformi i propri dati rispetto al modello di ingresso della Digital Library.

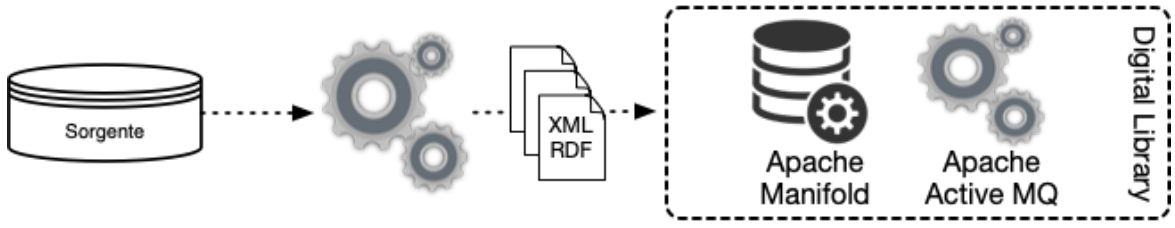

Figura 1: Scenario di riferimento

[1] https://manifoldcf.apache.org/
[2] https://activemq.apache.org/

3 Metodologia di Integrazione Dati

La Digital Library richiede che tutti i dati che si intendono integrare nel sistema debbano rispettare il formato di ingresso definito in [1] e risiedere nel database del sistema stesso. Quindi, affinchè una sorgente possa soddisfare queste condizioni deve implementare un processo che si occupi di ridurre l'eterogeneità sintattica e semantica tra i propri dati e il formato sintattico e semantico richiesto dalla Digital Library. Questo problema è riconducibile ai problemi noti in letteratura con il nome di Data (o Information o Knowledge) Integration o Extraction. Tra le le metodologie più efficaci per affrontare questo problema troviamo: *Extract Transform Load (ETL)* e Ontology-based Data Access (OBDA). Nei prossimi paragrafi verranno brevemente introdotte queste due metodologie. Fare riferimento a (Vassiliadis 2009) [7] per una completa descrizione della metodologia ETL, mentre una descrizione completa di OBDA si può fare riferimento a (Xiao et al. 2018)[8] e (Lenzerini 2002) [6].

Extract-Transform-Load (ETL). Il paradigma Extract-Transform-Load (ETL) prevede concettualmente tre fasi:

(i) *Extract.* La fase di estrazione si occupa di estrarre i dati dalle sorgenti omogenee o eterogenee;

(ii) *Transform.* La fase di trasformazione si occupa di trasformare i dati della sorgente riducendo l'eventuale eterogeneità sintattica e semantica tra la sorgente e il formato atteso (e.g. il modello di input della Digital Library);

(iii) *Load.* La fase di caricamento si occupa di caricare i dati trasformati nel sistema.

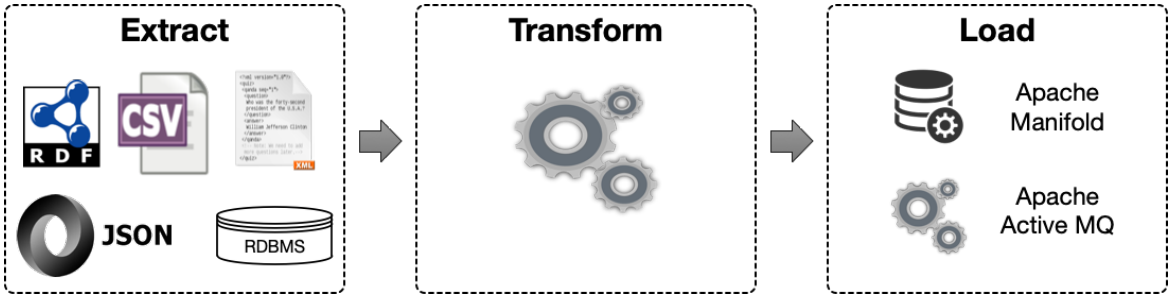

Figura 2: Il processo Extract-Transform-Load (ETL).

Ontology-based Data Access (OBDA). Ontology-based Data Access è un paradigma che permette di ottenere una vista omogenea di diverse sorgenti eterogenee. Il framework prevede tre elementi principali:

(i) *Ontologia*. L'ontologia è una rappresentazione formale ed esplicita di un certo dominio di interesse.

(ii) *Sorgente*. Lo schema dati della sorgente.

(iii) *Mapping*. Il mapping è un insieme di regole che definiscono come riscrivere le query definite secondo l'ontologia nei vocabolari adottati delle sorgenti.

La metodologia OBDA risulta essere molto più agile, in quanto per integrare una nuova sorgente secondo lo schema definito dall'ontologia basterà soltanto estendere il mapping. Tuttavia OBDA, diversamente da ETL, necessita che le sorgenti espongano un endpoint su cui eseguire le query.

Molte delle sorgenti censite nel task T3.1 (cf. Deliverable D3.1 [5]) non forniscono alcuna interfaccia che permetta di interrogare la risorsa. Quindi non si ritiene possibile adottare i paradigmi di integrazione dati che prevedono la virtualizzazione sorgenti (e.g. Ontology-based Data Access [8, 6]). Quindi, si ritiene che il paradigma più opportuno per l'integrazione delle sorgenti è Extract-Transform-Load (ETL) [7].

ETL per EcoDigit. Per quanto riguarda la prima e l'ultima fase (cioè estrazione e caricamento) è necessario seguire le modalità imposte dalla sorgente (per quanto riguarda l'attività di estrazione) e dalla Digital Library (per quanto riguarda l'attività di caricamento).

Per la fase di estrazione, si dovranno selezionare degli strumenti in grado di estrarre dati seguendo il formato imposto dalla sorgente. Come emerso dal Task 3.1 [5], i formati di esportazione più comuni sono quelli riconducibili ad un formato tabellare (e.g. CSV, Spreadsheet, SQL) o di markup (e.g. XML). Quindi la fase di estrazione dovrà permettere di trattare questa pletora di linguaggi.

La fase di caricamento, invece, consiste nel fornire il documento XML al componente di content management Apache Manifold o al gestore di code Apache Active MQ. Più precisamente, Apache Manifold gestisce i documenti in modalità *pull*, cioè permette di definire dei connettori che si occupano di prelevare i file da un repository. Apache Manifold offre una serie di connettori che permettono di configurare connessioni verso i più comuni repository come ad esempio servizi cloud (e.g. Google Drive, Dropbox etc.), dei database (e.g. Oracle, PostgreSQL etc.) o dei file system (e.g. HDFS). Apache Active MQ, invece, permette, a chi deve fornire i dati, di operare in modalità *push*. In questa modalità, è chi fornisce i dati a inviare il file XML a una coda di approvvigionamento. Quindi il risultato della fase di caricamento sarà la generazione di un file XML e il suo successivo caricamento nella Digital Library.

La fase di trasformazione prevede concettualmente a sua volta due sotto-fasi che hanno l'obiettivo di ridurre l'eventuale eterogeneità sintattica (la sorgente già potrebbe fornire i dati in XML) e l'eterogeneità semantica dei dati della sorgente con il formato atteso dalla Digital Library.

4 Strumenti per la Trasformazione Sintattica e Semantica dei Dati di una Sorgente

Per rendere conforme una sorgente alla Digital Library sono necessari, a seconda del suo formato sintattico e semantico, differenti strumenti e passi di trasformazione. Nelle prossime sezioni verranno passati in rassegna i principali strumenti che possono essere usati per rendere una sorgente conforme con la sintassi e semantica richiesta dalla Digital Library. La selezione degli strumenti presentati è stata guidata da un'analisi dello stato dell'arte in data integration e sull'analisi dei risultati del task di censimento [5]. Nel Deliverable D3.4 [2], presenteremo degli esempi concreti di come questi strumenti sono stati usati per trasformare i dati di alcune sorgenti selezionate.

Nella maggior parte dei casi, la trasformazione dei dati può essere effettuata in due fasi successive.

- *Prima Fase.* Nella prima viene ridotta l'eventuale eterogeneità *sintattica* tra il formato dati della sorgente e il formato atteso dalla Digital Library, provvedendo quindi a trasformare i dati in un linguaggio comune, cioè XML (formato adoattato dalla Digital Library).
- *Seconda Fase.* Nella seconda viene ridotta l'eterogeneità *semantica* tra i dati della sorgente e il formato atteso dalla Digital Library, provvedendo quindi a trasformare la struttura semantica dei dati (schema dei dati). Questa trasformazione può stata effettuata in due modi: o attraverso trasformazioni in linguaggio XML oppure contestualmente alla trasformazione sintattica della prima fase, se lo strumento lo permette.

Verrà inoltre mostrata una terza soluzione che, tramite la tecnica del templating permette di serializzare i dati provenienti direttamente da un applicativo software (senza dover passare attraverso serializzazione intermedie).

4.1 Strumenti per la Trasformazione Sintattica dei Dati

Gli strumenti per la trasformazione sintattica dei dati si occupano di trasformare i dati in un linguaggio comune, cioè XML (linguaggio adottato dalla Digital Library). Gli strumenti selezionati trattano i formati di dati più comuni: Database Relazionale (cf. Sezione 4.1.1), Sorgenti Tabellari (e.g. CSV, TSV, XSLX), YAML, JSON, RDF (in serializzazioni diverse da RDF-XML o tramite una interfaccia SPARQL).

4.1.1 Database Relazionale

Nel caso in cui una sorgente fornisca i dati permettendo l'accesso al proprio database relazionale è possibile usare D2RQ[3] [3] o Ontop[4] [4] per *(i)* permettere che i dati sia interrogati usando SPARQL; *(ii)* per esportare i dati in formato RDF/XML. In entrambi i casi questo tipo di strumenti necessitano di un file di mapping che indichi come trasformare i dati dal formato relazionale al formato RDF. Entrambi gli strumenti[5,6] accettano mapping in formato R2RML[7], formato standard W3C per esprimere mapping da database relazionali a RDF. E' importante notare che proprio grazie alla possibilità di definire dei mapping tra il formato relazionale e il modello RDF che è possibile formattare i dati della sorgente secondo il modello (semantico) atteso dalla Digital Library ed esportare i dati in formato RDF/XML (quindi effettuando contemporaneamente sia la trasformazione sintattica che quella semantica).

4.1.2 Formato Tabellare

Con formato tabellare si intende un qualsiasi formato che permetta l'esportazione di dati tabellari di una sorgente in file di testo (e.g. in caso di CSV e TSV) o in linguaggio XML (nel caso di XLSX). In questa sezione ci concentreremo sui formati tabellari per file di testo come CSV e TSV, in quanto il formato XSLX può essere: o processato come un XML, quindi necessiterà solo di una trasformazione semantica del contenuto (cf. Sezione 4.2), oppure può essere trasformato in CSV/TSV attraverso i più comuni strumenti per gestire fogli di calcolo. Nel caso in cui la sorgente esporti i dati in formato tabellare su file di testo (e.g. CSV e TSV) è possibile usare strumenti come Apache ANY23[8] o TARQL[9] per trasformare i dati in formato RDF/XML[10]. In entrambi i casi la trasformazione avviene in automatico producendo un file RDF/XML che rispecchia pedissequamente la struttura trasformando righe e colonne come segue:

(i) Ogni riga *i* corrisponde alla risorsa e_i del modello RDF;

(ii) Ogni colonna *j* corrisponde alla proprietà p_j del modello RDF;

(iii) Ogni cella *(i,j)* corrisponde al valore assegnato alla risorsa e_i per la proprietà p_j del modello RDF.

[3] http://d2rq.org/
[4] https://github.com/ontop/ontop
[5] http://d2rq.org/d2rq-language
[6] https://github.com/ontop/ontop/wiki/ObdalibRdb2rdf
[7] https://www.w3.org/TR/r2rml/
[8] https://any23.apache.org/
[9] http://tarql.github.io/
[10] TARQL permette anche di interrogare i dati usando SPARQL

4.1.3 YAML e JSON

YAML e JSON sono dei formati di serializzazione che vengono usati per scambiare dati tra sistemi software. Insieme a XML sono tra i più comuni formati di interscambio dati. Apache ANY23 può essere usato in entrambi i casi per trasformare i dati dal formato YAML o JSON al formato RDF/XML.

4.1.4 RDF e SPARQL

Nel caso in cui i dati siano disponibili in formato RDF ma con una serializzazione diversa da RDF/XML è possibile trasformare i dati della sorgente in questo formato usando una qualsiasi libreria tra le tante disponibili per la manipolazione di RDF (e.g. Apache Jena[11]).

Dallo studio dello stato dell'arte non sono emersi strumenti in grado di estrarre dati da uno SPARQL endpoint, frammentarli secondo una logica arbitraria (e.g. produrre un file per ogni risorsa) e serializzarli in RDF/XML. Dato che alcune sorgenti di interesse per il distretto (e.g. ArCo[12]) espongono i dati in SPARQL si è ritenuto necessario disporre di uno strumento che offra questo tipo di funzionalità.

Per questi motivi è stato sviluppato LOD-resource haverster (LOD-RH)[13]. LOD-RH è un nuovo strumento introdotto dal middleware Ecodigit che produce un file RDF/XML per ogni risorsa selezionata da uno SPARQL endpoint. LOD-RH prende come parametri la URL di uno SPARQL endpoint, una query per selezionare le risorse di interesse (i.e. una SELECT query) e una CONSTRUCT query che per ogni risorsa selezionata estragga i dati (i.e. graph pattern) di interesse per quella risorsa. Il sistema è basato su Apache Jena per l'interrogazione dello SPARQL endpoint e si può fare riferimento al repository GitHub per la documentazione aggiornata. Nel Deliverable D3.4 [2] mostreremo un esempio di uso di questo strumento per estrarre dati da una sorgente (i.e. ArCo). LOD-RH può essere usato sia per la trasformazione sintattica (i dati estratti dallo SPARQL endpoint vengono serializzati in RDF/XML) che per la trasformazione semantica (la query CONSTRUCT può essere usata anche per materializzare i dati direttamente nel formato atteso dalla Digital Library).

4.2 Strumenti per la Trasformazione Semantica dei Dati

Nella sezione 4.1 sono stati presentati gli strumenti che una sorgente può usare per trasformare i propri dati in RDF/XML. In questa sezione mostreremo come ristrutturare i dati (qualora ce ne fosse bisogno) seguendo il modello atteso dalla Digital Library (cf. D3.2 [1]). Le soluzioni per compiere questa attività sono principalmente due:

[11] https://jena.apache.org/
[12] http://dati.beniculturali.it/arco/
[13] https://github.com/ecodigit/lod-resource-harvester

(i) Se i dati sono in RDF/XML, la sorgente può caricare i dati in uno SPARQL endpoint e usare LOD-RH (cf. Sezione 4.1.4) per ristrutturare i dati secondo il modello atteso dalla Digital Library. E' importante notare che tool come D2RQ, Ontop o TARQL già permettono di interrogare i dati trasformati usando SPARQL. In altri casi è sempre possibile caricare i dati RDF/XML su uno SPARQL endpoint come Apache Jena Fuseki[14].

(ii) Nel caso in cui i dati della sorgente sono XML (in formato RDF o no), è possibile implementare delle trasformazioni usando il linguaggio XSLT che produca a partire un file RDF/XML a partire da un file XML della sorgente.

4.3 Strumenti di Templating

Il *templating* è una tecnica che permette di popolare un file (nel nostro caso in formato RDF/XML) con dati provenienti da un applicativo software. Il funzionamento di questi sistemi può essere descritto come segue.

Viene realizzato un documento di template avente la struttura sintattica e semantica desiderata (nel nostro caso quella attesa dalla Digital Library). Nel documento di template nei punti che dovrebbero contenere i valori reali legati alla risorsa descritta dal file si inseriscono delle variabili. Il nome delle variabili del template richiama il nome delle variabili del modello usato dall'applicativo software (e.g. nel caso in cui l'applicazione sia realizzata in Java le variabili vengono nominate secondo gli attributi delle classi Java che contengono i dati di interesse). Template e oggetto software contenente i dati viene passato ad un template engine che si occupa di popolare il template con i dati mantenuti nell'oggetto software. Questo meccanismo implementa il pattern software chiamato Model-View-Controller che permette di separare la presentazione dei dati dalla logica di business con cui sono ottenuti.

Apache FreeMarker[15] è una libreria Java che offre un template engine che accetta template definiti in linguaggio FTL e che si occupa di popolarli a partire da degli oggetti Java. La tecnica del templating evita, nel caso in cui i dati siano forniti da/estratti con un applicativo software, di dover serializzare i dati in un formato intermedio prima di dover procedere alla trasformazione. Nel Deliverable D3.4 [2] mostreremo come questa tecnica può essere usata per generare dati in modo compatibile con il modello di ingresso della Digital Library a partire dai dati raccolti tramite un software che permette di gestire i dati inseriti manualmente tramite form (e.g. Google Form).

4.4 Riepilogo

In questa sezione riepiloghiamo i risultati del censimento dei potenziali strumenti che una sorgente può adottare per rendere i propri dati conformi al modello di ingresso della Digital Library. Nella tabella 1 riportiamo una sintesi di questo censimento. Nella prima colonna

[14] `https://jena.apache.org/documentation/fuseki2/`
[15] `https://freemarker.apache.org/`

Formato Dati Sorgente	Strumenti per Trasformazione Sintattica in RDF/XML	Supporto a Trasformazione Semantica
RDBMS	D2RQ, Ontop	✓
Formato Tabellare (e.g. CSV)	Apache ANY23, TARQL	
YAML	Apache ANY23	
JSON	Apache ANY23	
RDF	Apache Jena	
SPARQL	LOD-RH	✓
XML	non necessario	Usando XSLT
Applicazione Software	Templating (e.g. Apache FreeMarker)	✓

Tabella 1: Sintesi del censimento degli strumenti.

della tabella viene riportato il formato sintattico di una potenziale sorgente, nella seconda gli strumenti che la potenziale sorgente può usare per ridurre l'eventuale eterogeneità sintattica con il modello di ingresso, nell'ultima colonna viene indicato se gli strumenti selezionati possono essere usati anche per trasformare la struttura semantica dei dati.

5 Conclusioni

In questo documento sono stati identificati potenziali strumenti concettuali, metodologici e tecnologici che una sorgente può adottare per rendere i propri dati conformi al modello di ingresso atteso dalla Digital Library e descritto nel Deliverable D3.2 [1]. In base alla tipologia delle sorgenti censite dal task 3.1 [5] si consiglia di adottare la metodologia di integrazione dati denominata Extract-Transform-Load, mentre nel caso in cui tutte lo sorgenti da integrare possano essere virtualizzate si consiglia di adottare la metodologia OBDA (Ontology-based Data Access). In seguito si è proceduto alla selezione di una serie di strumenti tecnologici che supportano queste due metodologie di integrazione dati e che possono essere usate dalla sorgente per rendere i propri dati conformi al modello di ingresso atteso dalla Digital Library.

Riferimenti bibliografici

[1] Luigi Asprino, Ludovica Marinucci, Andrea Giovanni Nuzzolese e Valentina Presutti. *D3.2 Modello di ingresso*. Deliverable Progetto EcoDigit. 2019.

[2] Luigi Asprino, Ludovica Marinucci, Andrea Giovanni Nuzzolese, Valentina Presutti, Massimo Mecella e Miguel Ceriani. *D3.4 Proof-of-Concept*. Deliverable Progetto EcoDigit. 2019.

[3] Christian Bizer e Andy Seaborne. "D2RQ-treating non-RDF databases as virtual RDF graphs". In: *Proceedings of the 3rd international semantic web conference (ISWC2004)*. Volume 2004. Proceedings of ISWC2004. 2004.

[4] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro e Guohui Xiao. "Ontop: Answering SPARQL queries over relational databases". In: *Semantic Web* 8.3 (2017), pagine 471–487.

[5] Miguel Ceriani e Massimo Mecella. *D3.1 Report sul Censimento*. Deliverable Progetto EcoDigit. 2019.

[6] Maurizio Lenzerini. "Data Integration: A Theoretical Perspective". In: *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*. A cura di Lucian Popa, Serge Abiteboul e Phokion G. Kolaitis. ACM, 2002, pagine 233–246. ISBN: 1-58113-507-6. DOI: 10.1145/543613.543644. URL: https://doi.org/10.1145/543613.543644.

[7] Panos Vassiliadis. "A Survey of Extract-Transform-Load Technology". In: *IJDWM* 5.3 (2009), pagine 1–27. DOI: 10.4018/jdwm.2009070101. URL: https://doi.org/10.4018/jdwm.2009070101.

[8] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati e Michael Zakharyaschev. "Ontology-Based Data Access: A Survey". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.* A cura di Jérôme Lang. ijcai.org, 2018, pagine 5511–5519. ISBN: 978-0-9992411-2-7. DOI: 10.24963/ijcai.2018/777. URL: https://doi.org/10.24963/ijcai.2018/777.