Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

ANALISI ESPLORATIVA E ARMONIZZAZIONE DI DATASET PER LA RAPPRESENTAZIONE GRAFICA MEDIANTE LA LIBRERIA D3

Tesi di Laurea in Tecnologie Web / Internet

Relatore:
Chiar.mo Prof.
FABIO VITALI

Presentata da:
PIERPAOLO ELIO JR DEL COCO

II Sessione
2012 - 2013

Indice

1 Introduzione

L'idea alla base della tesi è quella di sfruttare l'espressività di rappresentazione grafica della libreria D3 per mettere a punto un'applicazione web che dia la possibilità all'utente di creare dei grafici interattivi in maniera semplice e automatica. A tal proposito, si è subito considerato il dominio dell'applicazione, rappresentato dai dati da visualizzare in forma grafica, in modo da sviluppare un prodotto che, a differenza degli altri generatori di grafici, permetta l'elaborazione dei dataset reali, quelli cioè pubblicati sul web dai vari enti specializzati; un'applicazione che quindi indipendentemente dalla forma in cui i dati si presentano, riesca a rappresentarne il contenuto.

Nel corso degli ultimi anni, con l'avvento del movimento "Open Data" in risposta all'urlo di Sir. Tim Berners-Lee, "Raw Data, now!" [1], il web è più che mai un aggregatore di dati in continua crescita. Dalle agenzie governative, alle pubbliche amministrazioni, passando per gli istituti statistici e di ricerca, sono innumerevoli gli enti che scelgono il web per pubblicare i propri dati.

I dati in sé sono inutili, per questo sono detti "raw", grezzi; ma quando dai dati si estrapola il significato e lo si rappresenta, questi si trasformano in informazioni.

Da sempre il modo più veloce, efficace e affascinante di comunicare le informazioni contenute nei dati è costituito dalla visualizzazione grafica: un processo che interpreta i dati e li esprime sotto forma di proprietà visuali, molto più interessanti di un semplice insieme di numeri. Per rappresentare i dati possiamo avvalerci delle classiche visualizzazioni statiche, oppure utilizzare visualizzazioni interattive. Poiché le prime sono costituite da un'unica visuale sui dati, non rendono possibile la rappresentazione su più prospettive e quindi rappresentano un mezzo espressivo limitato; con le moderne tecnologie web è infatti possibile creare grafici dinamici

e interattivi, che spingono l'utente all'esplorazione dei dati e che diventano sempre più delle vere e proprie forme d'arte [2]. Inoltre creare grafici per il web, nel rispetto delle tecnologie standard dei browser ed evitando l'utilizzo di software proprietario o plug-in, è il modo migliore per raggiungere il massimo numero di utenti indipendentemente dal sistema operativo e dal tipo di device che questi ultimi utilizzano [21].
Le visualizzazioni interattive sul web combinano tecnologie diverse: HTML per il contenuto della pagina, CSS per la presentazione, JavaScript per l'interazione e SVG per i vettori grafici. Tutte queste tecnologie possono essere manipolate semplicemente attraverso un'unica libreria che prende il nome di D3, Data-Driven Documents, e che rappresenta uno strumento perfetto per la realizzazione di grafici interattivi per il web. D3 in realtà non è un framework di visualizzazione tradizionale: piuttosto che offrire un elenco chiuso di grafici preconfezionati, mette a disposizione un vero e proprio "kernel" di visualizzazione il cui scopo è la trasformazione degli elementi di una pagina web in base ai dati [16]. Questo rende D3 uno strumento potentissimo, la cui logica è molto più vicina ai trasformatori di documenti come jQuery, CSS e XSLT, invece che alle comuni librerie di grafici.

Come accennato precedentemente nel capitolo, il numero di enti specializzati nella pubblicazione di dati sul web è in continua crescita: come esempi a livello internazionale si possono citare World Bank, Eurostat, World Health Organizzation; mentre a livello nazionale l'Istat, ma anche tutte le pubbliche amministrazioni che hanno aderito agli Open Data; e ognuno di questi enti pubblica i dati secondo un proprio modello. Per questo motivo, nella realizzazione dell'applicazione descritta lo scoglio principale è rappresentato dall'eterogeneità dei dataset provenienti da fonti diverse.
Il prototipo sviluppato prende il nome di DAD3, Dataset Adapter to D3, poiché il cuore del progetto si occupa fondamentalmente di risolvere il problema dell'eterogeneità, collocandosi tra il dataset e il grafico, adattando appunto la forma in cui si presentano i dati all'interfaccia di visualizzazione grafica. Attraverso un'analisi esplorativa dei dati, DAD3 ne deduce la natura, individua tra tutte le informazioni le sole utili alla visualizzazione grafica, scartando invece quelle relative alla struttura del dataset e infine a sceglie la forma di rappresentazione migliore per i dati analizzati.

La tesi si articola in sei capitoli in cui si illustrano le motivazioni che hanno portato alla costruzione del progetto, di cui poi se ne descrive il funzionamento e i dettagli implementativi; successivamente si valuta il prototipo, in base all'efficienza e al suo effettivo contributo alla risoluzione del problema, motivo della sua realizzazione. Infine si riportano le funzionalità aggiuntive da sviluppare in futuro per concretizzare le prospettive aperte dalla prima versione del prototipo.

Nel capitolo 2 si descrive il problema dell'eterogeneità dei dataset reali e come esso rappresenti lo scoglio principale nella costruzione di un sistema per la rappresentazione automatica di grafici a partire dai dataset pubblicati sul web dalle diverse organizzazioni. Nel dettaglio si analizzano due dataset provenienti da Eurostat e World Bank, che pur essendo strutturati in modo completamente diverso, esprimono fondamentalmente un concetto molto simile che dev'essere rappresentato graficamente nello stesso modo.
Successivamente si espongono alcune delle soluzioni presenti in letteratura e sul mercato in risposta al medesimo problema.

Nel capitolo 3 si illustra il progetto di tesi DAD3 come soluzione al problema dell'eterogeneità dei dataset: si descrive il principio su cui si basa il progetto, adattare cioè i dataset all'interfaccia di visualizzazione dei grafici in modo da garantirne l'interoperabilità. Nel corso del capitolo si evidenzia come DAD3, Dataset Adapter to D3, si collochi all'interno del pattern strutturale adapter, da cui ne trae il nome. Successivamente, da una prima descrizione generale del funzionamento di DAD3, si passa all'analisi di un caso concreto, che dimostra come DAD3 riesca a dedurre la natura dei dati contenuti nei due dataset precedentemente illustrati e come riesca a rappresentarli graficamente nel modo migliore.

Nel capitolo 4 si analizza dettagliatamente l'implementazione del progetto DAD3. Da una prima introduzione alla libreria D3, in cui si mette in evidenza l'espressività di questo strumento per la creazione di grafici interattivi per il web, si passa alla descrizione dell'architettura di DAD3, spiegando il funzionamento di ogni modulo che la compone. Di conseguenza si delineano le varie fasi che entrano in gioco nell'analisi, l'estrapolazione e l'armonizzazione dei dati, illustrandole nello stesso ordine con cui sono eseguite dal software. Si prende quindi in esame innanzitutto

la fase di lettura del sorgente XML e le tecnologie utilizzate per incrementarne l'efficienza. Si passa poi a esporre l'analisi lessicale, fase in cui si distinguono i dati quantitativi e i dati qualitativi rilevati nel corso della lettura del documento assegnando loro un tipo; per poi proseguire con la fase di analisi semantica, in cui si spiega come DAD3 deduca il significato che i dati assumono all'intero del documento sorgente. A questo punto si descrive il modo in cui i dati quantitativi e i relativi dati qualitativi sono estrapolati dal dataset per essere prima memorizzati in nuovi record dalla struttura lineare e in seguito privati dalle informazioni inutili alla rappresentazione grafica: si spiega dunque come i dataset siano armonizzati, rifiniti e adattatati all'interfaccia di visualizzazione.

Nel capitolo 5 si valuta il progetto DAD3 sia dal punto di vista dell'efficienza, sia dal punto di vista della qualità dei risultati ottenuti, descrivendone sia i pregi che i limiti.
Infine nel capitolo 6 si riprende il concetto di come DAD3 rappresenti, alla luce dei risultati ottenuti, una soluzione valida al problema dell'eterogeneità dei dataset reali, poiché consente l'utilizzo di quest'ultimi in un sistema di generazione automatica di grafici. Inoltre si espone, alla fine del capitolo, un elenco di nuove funzionalità da sviluppare in futuro per garantire una migliore usabilità ed espressività del progetto DAD3.

2 | L'eterogeneità dei Dataset

I dataset disponibili sul world wide web presentano due problemi fondamentali:

- problemi di natura **locale**: come l'accessibilità dei dati da parte delle applicazioni e la scarsa disponibilità per un determinato formato sia esso XML, JSON, CSV, ecc... ;

- problemi di natura **globale**: come l'eterogeneità dell'insieme dei dataset provenienti da diverse fonti.

Il lavoro di questa tesi mira a proporre una soluzione al problema dell'eterogeneità dei dataset.
Poiché ogni organizzazione pubblica i dataset seguendo un proprio modello, ne risulta un insieme tanto vasto quanto eterogeneo, che non consente l'interoperabilità tra le applicazioni che li visualizzano graficamente. Questo comporta di fatto la modifica in tutto o in parte dell'applicazione grafica per ogni diversa fonte di dati utilizzata, anche quando questi ultimi, pur presentandosi in forme diverse, esprimino un concetto del tutto simile. Il programmatore deve adattare manualmente i dataset all'interfaccia della sua applicazione. Ovviamente questo comporta un enorme ostacolo nella realizzazione di sistemi che permettano all'utente di generare automaticamente grafici interattivi a partire da dataset a sua scelta. In figura 2.1 e in figura 2.2, sono mostrate due diverse sorgenti di dati in XML, rispettivamente Eurostat [3] e World Bank [4], dalla forma completamente diversa, ma fondamentalmente contenenti dati della stessa natura: entrambi i dataset descrivono l'andamento temporale di un valore numerico, la popolazione in un caso e il GDP nell'altro, espresso per diversi paesi.

```
<Data>
   <Table>
      <Grid>
         <AxisY name="geo">
            <Position value="EU28">...</Position>
               ...
            <Position value="FR">...</Position>
            <Position value="HR">...</Position>
            <Position value="IT">
               <AttList>
                  <Att name="order">16</Att>
               </AttList>
               <AxisX name="time">
                  <Position value="2002">...</Position>
                  <Position value="2003">...</Position>
                  <Position value="2004">...</Position>
                  <Position value="2005">...</Position>
                  <Position value="2006">...</Position>
                  <Position value="2007">...</Position>
                  <Position value="2008">...</Position>
                  <Position value="2009">...</Position>
                  <Position value="2010">
                     <AttList>
                        <Att name="order">4</Att>
                     </AttList>
                     <Cell value="60340328"/>
                  </Position>
                  <Position value="2011">
                     <AttList>
                        <Att name="order">3</Att>
                     </AttList>
                     <Cell value="60626442"/>
                  </Position>
                  <Position value="2012">
                     <AttList>
                        <Att name="order">2</Att>
                     </AttList>
                     <Cell value="59394207">
                        <Ref value="bp"/>
                     </Cell>
                  </Position>
                  <Position value="2013">...</Position>
               </AxisX>
            </Position>
```

```
            <Position value="CY">...</Position>
                ...
            <Position value="BA">...</Position>
            <Position value="XK">...</Position>
         </AxisY>
      </Grid>
      <Information>
         <Group type="context">...</Group>
         <Group type="nomenclature">
            <Group type="language" value="en">
               <Group type="dictionary" value="context">...</Group>
               <Group type="dictionary" value="dimension">...</Group>
               <Group type="dictionary" value="indic_de">...</Group>
               <Group type="dictionary" value="dimension">...</Group>
               <Group type="dictionary" value="geo">...</Group>
               <Group type="dictionary" value="dimension">...</Group>
               <Group type="dictionary" value="time">...</Group>
               <Group type="dictionary" value="flagsfootnotes">...</Group>
            </Group>
            <Group type="language" value="fr">...</Group>
            <Group type="language" value="de">...</Group>
         </Group>
      </Information>
   </Table>
</Data>
```

Figura 2.1: Popolazione per Stato e per anno: Italia 2010, 2011 e 2012 in evidenza. Fonte Eurostat.

```
<wb:data xmlns:wb="http://www.worldbank.org" page="1" pages="1"
per_page="50" total="15">
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="AT">Austria</wb:country>
      <wb:date>2012</wb:date>
      <wb:value>399649131196.966</wb:value>
      <wb:decimal>0</wb:decimal>
   </wb:data>
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="AT">Austria</wb:country>
      <wb:date>2011</wb:date>
      <wb:value>417656162500</wb:value>
      <wb:decimal>0</wb:decimal>
   </wb:data>
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="AT">Austria</wb:country>
      <wb:date>2010</wb:date>
      <wb:value>376837981578.947</wb:value>
      <wb:decimal>0</wb:decimal>
   </wb:data>
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="DE">Germany</wb:country>
      <wb:date>2012</wb:date>
      <wb:value>3399588583183.34</wb:value>
      <wb:decimal>0</wb:decimal>
   </wb:data>
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="DE">Germany</wb:country>
      <wb:date>2011</wb:date>
      <wb:value>3600833333333.33</wb:value>
      <wb:decimal>0</wb:decimal>
   </wb:data>
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="DE">Germany</wb:country>
      <wb:date>2010</wb:date>
      <wb:value>3284473684210.53</wb:value>
      <wb:decimal>0</wb:decimal>
   </wb:data>
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="ES">Spain</wb:country>
      <wb:date>2012</wb:date>
      <wb:value>1349350732836.2</wb:value>
      <wb:decimal>0</wb:decimal>
   </wb:data>
   <wb:data>
      <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
      <wb:country id="ES">Spain</wb:country>
      <wb:date>2011</wb:date>
      <wb:value>1476881944444.44</wb:value>
      <wb:decimal>0</wb:decimal>
</wb:data>
```

```
    <wb:data>
        <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
        <wb:country id="ES">Spain</wb:country>
        <wb:date>2010</wb:date>
        <wb:value>1380109210526.32</wb:value>
        <wb:decimal>0</wb:decimal>
    </wb:data>
    <wb:data>
        <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
        <wb:country id="FR">France</wb:country>
        <wb:date>2012</wb:date>
        <wb:value>2612878387760.35</wb:value>
        <wb:decimal>0</wb:decimal>
    </wb:data>
    <wb:data>
        <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
        <wb:country id="FR">France</wb:country>
        <wb:date>2011</wb:date>
        <wb:value>2779719500000</wb:value>
        <wb:decimal>0</wb:decimal>
    </wb:data>
    <wb:data>
        <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
        <wb:country id="FR">France</wb:country>
        <wb:date>2010</wb:date>
        <wb:value>2548315434210.53</wb:value>
        <wb:decimal>0</wb:decimal>
    </wb:data>
    <wb:data>
        <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
        <wb:country id="IT">Italy</wb:country>
        <wb:date>2012</wb:date>
        <wb:value>2013263114238.88</wb:value>
        <wb:decimal>0</wb:decimal>
    </wb:data>
    <wb:data>
        <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
        <wb:country id="IT">Italy</wb:country>
        <wb:date>2011</wb:date>
        <wb:value>2192357094734.72</wb:value>
        <wb:decimal>0</wb:decimal>
    </wb:data>
    <wb:data>
        <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
        <wb:country id="IT">Italy</wb:country>
        <wb:date>2010</wb:date>
        <wb:value>2041954747600</wb:value>
        <wb:decimal>0</wb:decimal>
    </wb:data>
</wb:data>
```

Figura 2.2: GDP in $ per Stato e per anno: Austria, Italia, Germania, Francia e Spagna; 2010, 2011 e 2012. Fonte World Bank

Come si può notare entrambi i dataset sono caratterizzati da uno schema proprio; inoltre sono presenti informazioni inutili e fuorvianti rispetto al concetto da rappresentare graficamente.

2.1 Una soluzione: l'uso degli Standard

Una possibile soluzione del problema è la proposta di uno standard nella costruzione dei dataset. Un esempio è dato da **SDMX**. SDMX, **Statistical Data and Metadata Exchange**, [5] rappresenta un'iniziativa di cooperazione internazionale per lo sviluppo e l'impiego di processi più efficienti per lo scambio e la condivisione di dati e metadati statistici tra le organizzazioni internazionali più importanti. L'iniziativa, avviata nel 2001, ha lo scopo di stabilire una serie di standard riconosciuti e osservati da tutti gli operatori, per facilitare l'accesso ai dati statistici e soprattuto per garantire che i dati siano sempre associati a metadati che attribuiscono loro significato, utilità e comprensibilità da parte delle applicazioni. Di seguito viene illustrato un esempio di formalizzazione in XML di questo standard: SDMX rappresenta una soluzione interessante al problema dell'eterogeneità per-

```
<org:DataSet>
   <org:Series FREQUENCY="A" REFERENCE_AREA="CH" TOPIC="03">
      <org:Obs TIME="2004" OBSERVATION_VALUE="3.145"/>
      <org:Obs TIME="2005" OBSERVATION_VALUE="2.96"/>
      <org:Obs TIME="2006" OBSERVATION_VALUE="3.457"/>
      <org:Obs TIME="2007" OBSERVATION_VALUE="4.206"/>
   </org:Series>
</org:DataSet>
```

Figura 2.3: Esempio di frammento SDMX in XML.

ché pone l'accento non solo sull'utilizzo dei metadati all'interno dei dataset, ma anche sull'armonizzazione delle strutture dei dati [19]. I dataset pubblicati seguendo le specifiche dettate da questo standard risultano omogenei e permettono alle applicazioni grafiche di interoperare con tutte le sorgenti di dati che usano SDMX. Purtroppo però questo stardard può essere associato solo all'utilizzo di un linguaggio di scambio di dati come XML. Non si può dunque trascurare il fatto che nella realtà esiste un vasto insieme di dataset contenenti dati interessanti, non

tutti strutturati in forma standard e non tutti scritti in XML. Anche se il prototipo DAD3 in questa sua prima versione considera i dataset scritti in XML, occorre trovare una soluzione generale che fornisca risultati interessanti indipendentemente dal formato di interscambio dei dati.

2.2 Ulteriori soluzioni

Di seguito si riportano due soluzioni che rappresentano strumenti molto interessanti per l'interoperabilità dei dati.

2.2.1 OpenRefine

OpenRefine [6], ex progetto di Google denominato Google Refine, è uno strumento essenziale per ripulire collezioni di dati, che spesso contengono informazioni aggiuntive inutili o non omogenee, prima di procedere alla loro visualizzazione. Per esempio se si hanno dati da siti, file di testo, fogli di calcolo, che usano categorie diverse per riferirsi ad una stessa tipologia di informazione (es. il sesso può essere definito come maschi/femmine o M/F in tabelle diverse) si può decidere quale convenzione usare ed estenderla ai valori non omogenei. In questo modo si otterrà un unico dataset ordinato.

2.2.2 Google Fusion Tables

Google Fusion Tables [17] è un progetto cloud-based nato nel 2009 come un ambiente di collaborazione per gli utenti che lavorano con i dati. Il suo obiettivo è quello di offrire nuove funzionalità per la gestione e la condivisione di grandi dataset e soprattutto permettere la fusione delle tabelle di dati pubblicate da utenti diversi. Il progetto è interessante poiché senza richiedere informazioni sul tipo di dati caricati dall'utente riesce a indovinare la natura dei dati, permettendo quindi di importare un dataset in maniera molto semplice e direttamente dalla fonte da cui lo si è prelevato senza doverlo prima adattare. Inoltre il sistema permette di creare altrettanto facilmente grafici con i dati importati. Per questo motivo Google Fusion Tables rappresenta un'ottimo strumento per la gestione e la condivisione dei dati, basato interamente sull'interoperabilità dei diversi dataset. Del

resto Google Fusion Tables, pur essendo un eccellente analizzatore di dati, accetta i formati csv, tsv e txt, ma non ad esempio json o xml; inoltre trattandosi di un sistema volutamente aperto ai soli collaboratori, non consente l'inclusione dei grafici generati su una pagina web esterna, quale potrebbe essere ad esempio l'esigenza di un blogger che vuole condividere il suo grafico sul blog.

3 | DAD3: Dataset Adapter to D3

A causa dell'eterogeneità dei dataset, come spiegato nel capitolo 2, le rappresentazioni grafiche dei dati spesso non sono interoperabili: è compito del programmatore sviluppare, oltre al grafico, un modulo tra l'acquisizione dei dati e la visualizzazione per ogni diversa sorgente di dati da rappresentare. Nei casi peggiori, in cui i dati sono mischiati alla logica rappresentativa, si ha un'applicazione costruita specificatamente per un'unica fonte di dati. Avere una struttura omogenea dei dataset semplifica lo sviluppo di grafici web interattivi e consente il riutilizzo di uno stesso grafico per rappresentare più dataset contenenti le stesse unità statistiche ma dalla struttura completamente differente. Dal punto di vista del progetto di tesi, disporre di un software che adatti i dataset all'interfaccia di visualizzazione rappresenta la base su cui costruire un sistema automatico per la rappresentazione grafica dei dati.

3.1 Cos'è DAD3

Esistono tantissimi lavori sulla generazione automatica di grafici [22] [20] [7], alcuni dei quali analizzano i dataset in ingresso automaticamente, ma nessuno di essi offre una flessibilità così elevata nell'analisi automatica dei dataset come invece fa DAD3. Il progetto DAD3 è più simile a strumenti come OpenRefine e Google Fusion Table, ma si spinge oltre: al pari di OpenRefine offre infatti la possibilità di ripulire i dati da informazioni inutili, ma lo fa automaticamente; inoltre, come Google Fusion Table, offre un analizzatore automatico dei dati, che anche se in questa versione accetta solo dataset in XML, è predisposto per accettare in futuro anche altri formati diffusi.

DAD3, Dataset Adapter to D3, è un'applicazione web che adatta automaticamente i dataset reali all'interfaccia della libreria D3, oltre che proporre una rappresentazione grafica appropriata alla natura dei dati. DAD3 offre dunque una soluzione al problema dell'interoperabilità dei grafici in D3, consentendo il riuso di modelli già implementati senza dover modificare il codice e rende quindi possibile la generazione automatica di grafici per la rappresentazione di dataset eterogenei.

A partire dal dataset caricato, l'applicazione consente di selezionare tramite un'interfaccia quali categorie e unità statistiche visualizzare nel grafico, mostrando le modifiche immediatamente.

Il progetto nasce come parte di un sistema per la generazione automatica di grafici da parte degli utenti, ma si presta anche all'utilizzo da parte dei programmatori poiché è prevista la possibilità di esportare il dataset armonizzato per D3, permettendo così di utilizzare il dataset riadattato per creare una propria applicazione grafica in D3.

3.2 DAD3 nel pattern Adapter

Il progetto DAD3 prende posto come partecipante nel pattern Adapter [23], da cui appunto prende il nome. L'adapter è un pattern strutturale che ha il compito di fornire una soluzione astratta al problema dell'interoperabilità tra interfacce differenti. La sua struttura è mostrata in figura 3.1 e comprende i partecipanti:

- **Adaptee**: definisce l'interfaccia che ha bisogno di essere adattata;
- **Target**: definisce l'interfaccia utilizzata dall'utente;
- **Adapter**: adatta l'interfaccia Adaptee all'interfaccia Target.

Nel caso specifico l'Adaptee è costituito da un generico dataset che dev'essere adattato all'interfaccia del grafico, rappresentato dal partecipante Target. L'architettura di DAD3 è ripresa nel capitolo 4, dove lo si analizzerà in maniera più dettagliata.

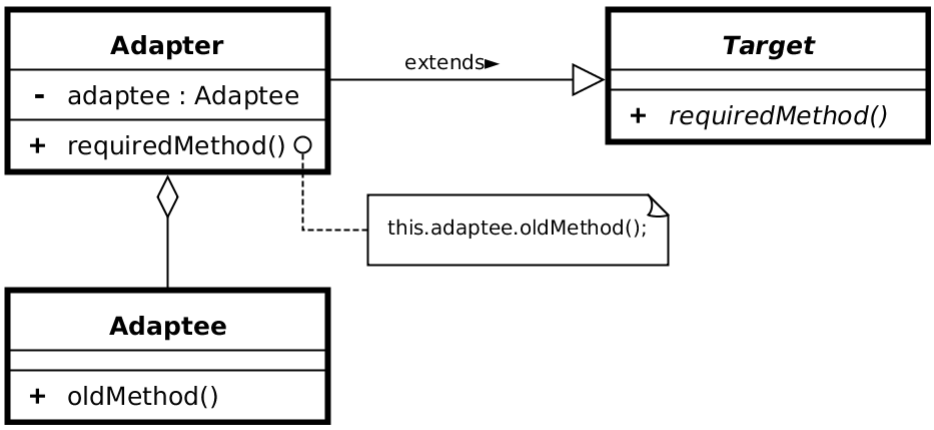

Figura 3.1: Diagramma UML del pattern Adapter: fonte Wikipedia.org [8].

3.3 Analisi esplorativa e armonizzazione

DAD3 effettua un'analisi esplorativa di dataset di cui a priori non conosce ne il contenuto, ne lo schema; tenta cioè di dedurre la natura dei dati, il loro significato all'interno del dataset e la loro struttura per scegliere infine la forma più appropriata di visualizzazione.
I dati quantitativi e i rispettivi dati qualitativi sono estrapolati dal dataset sorgente per poi essere armonizzati, cioè riassemblati in nuovi record dalla struttura nota e indipendente dal sorgente, dunque adattati all'interfaccia di visualizzazione.
Prima di passare i dati armonizzati al modulo di visualizzazione, DAD3 li rifinisce eliminando tutte le informazioni superflue, in modo che i nuovi record contengano tutte e sole le informazioni necessarie alla rappresentazione grafica.

3.4 La semantica dei dati

Oltre ad assumere una struttura più semplice e lineare, i dati armonizzati a questo punto hanno acquisito valore semantico: le date, dapprima semplici stringhe, ora oggetti JavaScript di tipo Date; i nomi geografici, unità statistiche collegate ad oggetti TopoJSON [15] [9] rappresentabili graficamente tramite le cartografie associate; e infine i numeri, sono tutti dati associati a delle entità.

Anche se allo stato attuale DAD3 può attribuire significato solo a questi tre tipi di dato, questo risultato è molto importante poiché predispone il sistema alla possibilità di intrecciare i dati provenienti da due o più sorgenti diverse se i dati in esse contenuti descrivono le stesse unità statistiche.

3.5 La rappresentazione grafica

Dopo aver analizzato e armonizzato i dati di un dataset, DAD3 sceglie la rappresentazione grafica più appropriata alla loro natura. Al momento i dati possono essere rappresentati mediante tre grafici diversi: a linee, a barre e a dispersione.

3.6 Un caso concreto

Per comprendere meglio il funzionamento di DAD3 è utile analizzare un esempio concreto.
Si considerino due estratti dei dataset riportati nel capitolo 2 provenienti da World Bank e Eurostat, rispettivamente in figura 3.2 e 3.3. Questi due dataset, come evidenziato nel capitolo 2, hanno una struttura completamente diversa, ma condividono la stessa natura: per ogni paese, carattere dell'osservazione, esiste un valore numerico che varia al variare del tempo; in un caso questo valore è costituito dalla popolazione, mentre nell'altro dal GDP espresso in dollari. Momentaneamente si considerino solo i caratteri *IT*, per il dataset Eurostat, e *Italy*, per il dataset World Bank, entrambi per gli anni *2010*, *2011* e *2012*. Per comodità questi estratti dei precedenti dataset sono mostrati nelle figure 3.2 e 3.3.

```
</wb:data>
  ...
  <wb:data>
    <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
    <wb:country id="IT">Italy</wb:country>
    <wb:date>2012</wb:date>
    <wb:value>2013263114238.88</wb:value>
    <wb:decimal>0</wb:decimal>
  </wb:data>
```

```
  <wb:data>
    <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
    <wb:country id="IT">Italy</wb:country>
    <wb:date>2011</wb:date>
    <wb:value>2192357094734.72</wb:value>
    <wb:decimal>0</wb:decimal>
  </wb:data>
  <wb:data>
    <wb:indicator id="NY.GDP.MKTP.CD">GDP (current US$)</wb:indicator>
    <wb:country id="IT">Italy</wb:country>
    <wb:date>2010</wb:date>
    <wb:value>2041954747600</wb:value>
    <wb:decimal>0</wb:decimal>
  </wb:data>
  ...
</wb:data>
```

Figura 3.2: Estratto del dataset in figura 2.2

```
<Position value="IT">
  ...
  <Position value="2010">
    <AttList>
      <Att name="order">4</Att>
    </AttList>
    <Cell value="60340328">
      <Ref value="p"/>
    </Cell>
  </Position>
  <Position value="2011">
    <AttList>
      <Att name="order">3</Att>
    </AttList>
    <Cell value="60626442">
      <Ref value="p"/>
    </Cell>
  </Position>
  <Position value="2012">
    <AttList>
      <Att name="order">2</Att>
    </AttList>
    <Cell value="60820696">
      <Ref value="p"/>
    </Cell>
  </Position>
</Position>
```

Figura 3.3: Estratto del dataset in figura 2.1

Dopo l'analisi, l'armonizzazione e la rifinitura i dataset appariranno come mostrato nelle seguenti tabelle:

(a) Armonizzazione del dataset 3.3

Key	**Value**	**Date**
...	...	...
IT	60820696	2012
IT	60626442	2011
IT	60340328	2010
...	...	...

(b) Armonizzazione del dataset 3.2

Key	**Value**	**Date**
...	...	...
Italy	2013263114238.88	2012
Italy	2192357094734.72	2011
Italy	2041954747600	2010
...	...	...

Tabella 3.1: Dataset armonizzati

Anche la forma tabellare dei nuovi dataset è, per motivi di spazio, solo un estratto; infatti nei dataset armonizzati seguono sequenzialmente alle unità mostrate tutte le altre unità rilevate dai sorgenti, con i realitivi valori numerici e le relative date. Come si può notare, entrambi i dataset ora sono caratterizzati dalla stessa struttura: il valore numerico dell'osservazione è memorizzato nel campo **value**, è presente un campo **date** in cui è memorizzato l'anno e il campo **key** contiene l'unità statistica. I caratteri *IT* e *Italy* pur essendo due stringhe diverse sono entrambi nomi geografici che puntano allo stesso oggetto TopoJSON nel database, predisposto per una visualizzazione cartografica.
A questo punto selezionando tramite l'interfaccia solo alcuni caratteri in un intervallo di tempo più ampio, al fine di ottenere una rappresentazione più comprensibile, si ottengono i seguenti grafici:

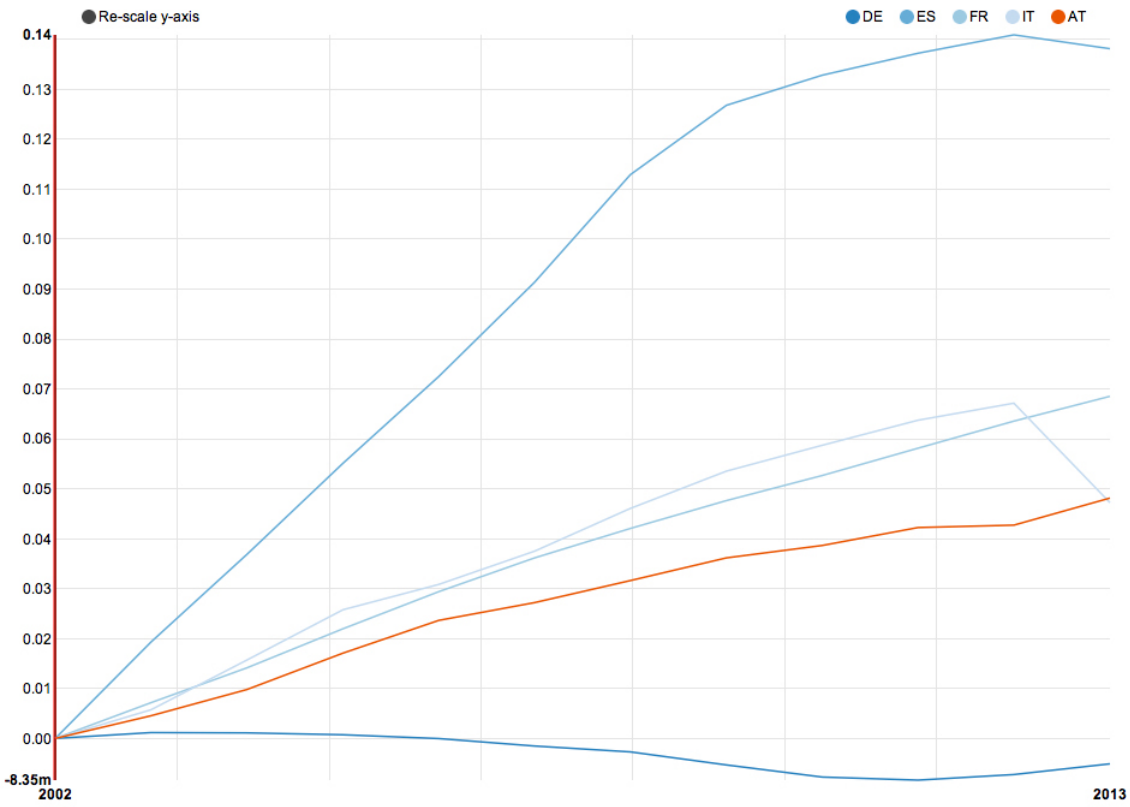

Figura 3.4: Grafico per il dataset in figura 2.1

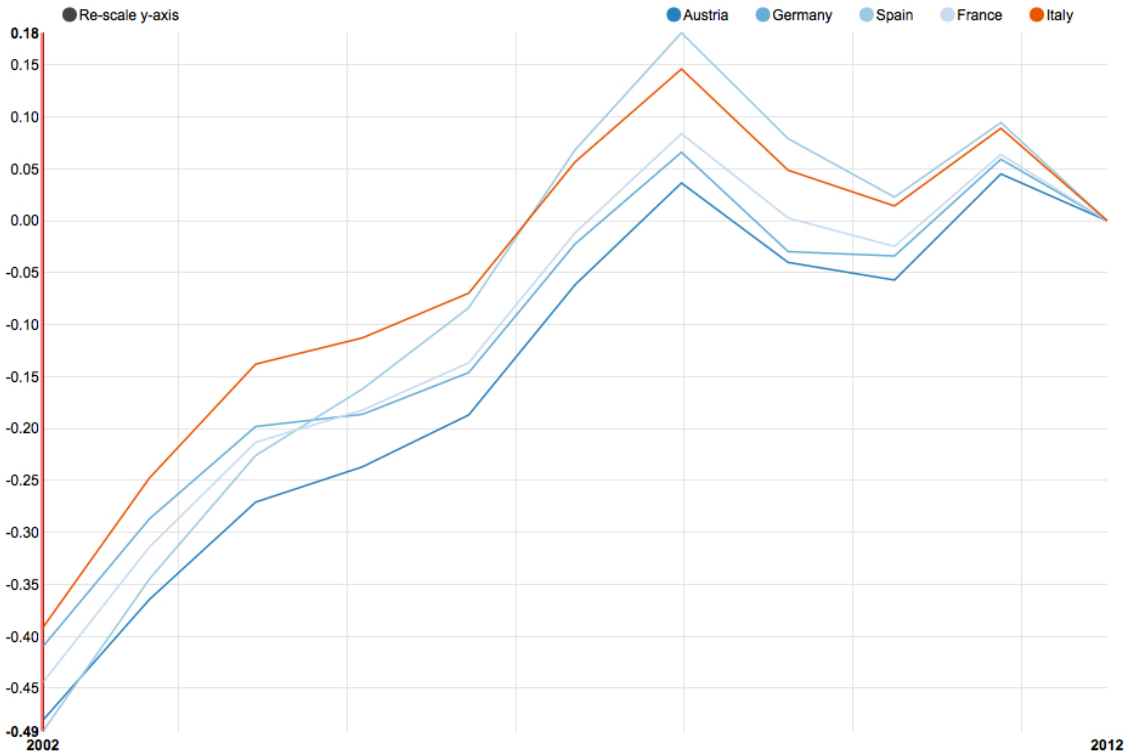

Figura 3.5: Grafico per il dataset in figura 2.2

Come si può vedere dai grafici in figura 3.4 e 3.5, il concetto espresso dai due dataset sorgenti è stato rappresentato nello stesso modo pur avendo questi delle strutture completamente differenti.

4 | Implementazione di DAD3

Prima di passare alla descrizione del progetto DAD3 e della sua architettura, è utile introdurre D3, la libreria di grafici alla base del sistema implementato.

4.1 Un'introduzione a D3

D3 (Data Driven Documents) [16] [10] è una libreria Javascript scritta da Mike Bostock come progetto successore di un precedente tool di visualizzazione chiamato Protovis. E' basata sugli standard web e sfrutta appieno le tecnologie dei browser per manipolare gli elementi: come per JQuery, si utilizza la sintassi CSS per i selettori e si applicano gli stili agli elementi tramite fogli CSS. D3 a differenza delle altre librerie di grafici, non offre un insieme di grafici già pronti all'uso, bensì un potente framework che permette di realizzare praticamente qualsiasi tipo di grafico manipolando gli elementi di una pagina web di tipo HTML, SVG o Canvas in base al contenuto di un dataset. Di seguito è mostrato un piccolo esempio di grafico e il relativo codice in D3.

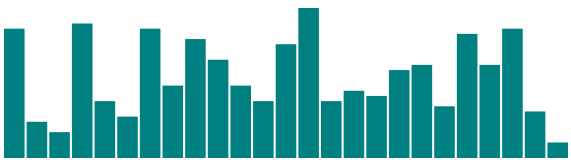

Figura 4.1: Un semplice grafico a barre con D3 [11]

```
var dataset = [ 25, 7, 5, 26, 11, 8, 25, 14, 23, 19,
                14, 11, 22, 29, 11, 13, 12, 17, 18, 10,
                24, 18, 25, 9, 3 ];

d3.select("body").selectAll("div")
    .data(dataset)
    .enter()
    .append("div")
    .attr("class", "bar")
    .style("height", function(d) {
        var barHeight = d * 5;
        return barHeight + "px";
    });
```

Listing 4.1: Un frammento di codice in D3 [11]

Come si può notare, il cuore dello script è rappresentato dalla funzione **data()**, la quale scorre il dataset per tutta la sua lunghezza passando ogni dato ai metodi richiamati in sequenza; così per ogni dato viene creato un elemento *div* nella pagina a cui sono applicati degli stili coerentemente con il dato associato. Quando tramite la funzione **enter()** si entra nel dato corrente, non si dispone più di una visione globale del dataset; per questo è molto importante che ogni dato riporti tutte le informazioni utili alla rappresentazione grafica, in maniera ridondante rispetto agli altri dati. Per utilizzare questa libreria in tutta la sua espressività, occorre un modo per trasformare i dataset reali, dalla struttura complessa e ricchi di informazioni superflue, in dataset simili a quello dell'esempio.

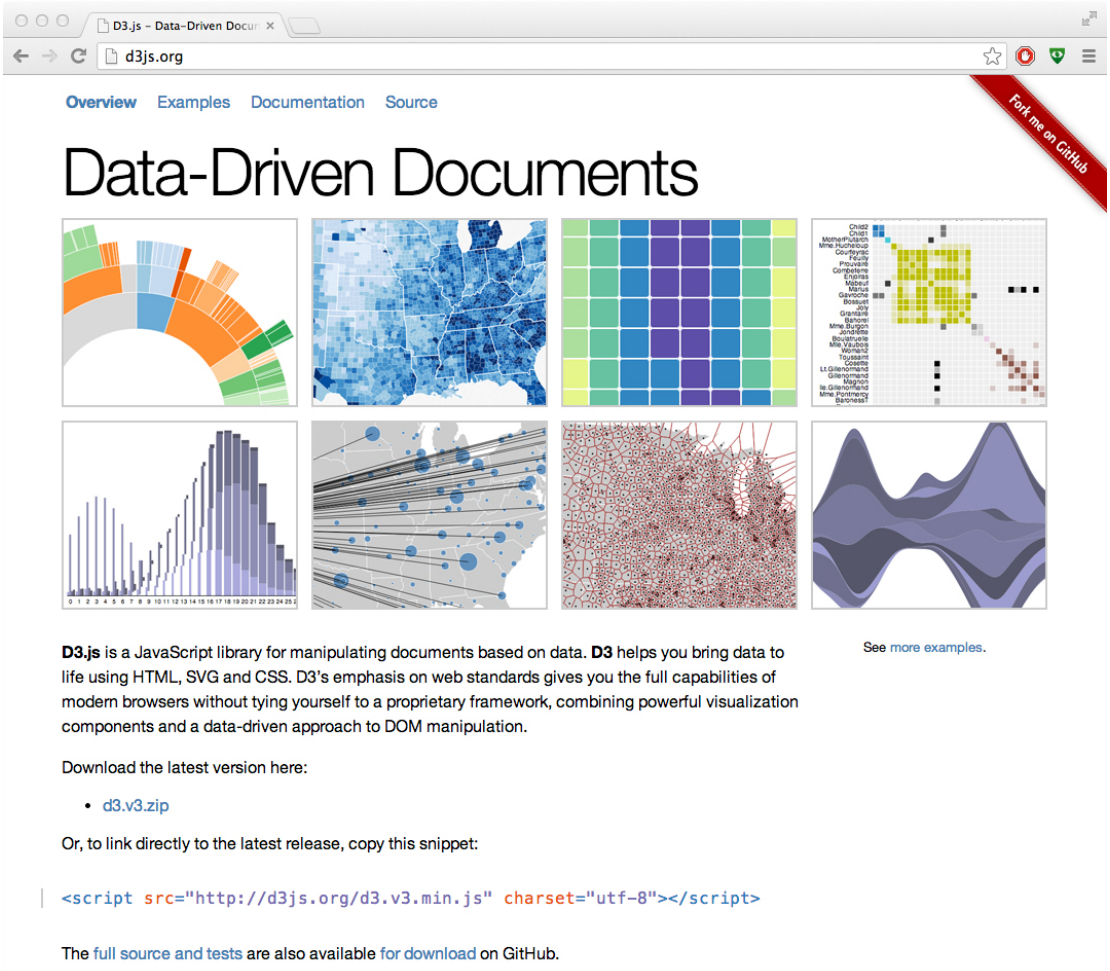

Figura 4.2: Sito del progetto D3 [10]

4.2 Architettura di DAD3

La figura 4.3 mostra l'architettura del progetto basata sul pattern strutturale adapter, il cui fine è quello di fornire una soluzione astratta al problema dell'interoperabilità tra interfacce differenti. Nello specifico i dati presenti nel documento XML devono essere adattati, quindi armonizzati e ristrutturati in un array per l'utilizzo da parte della libreria di grafici D3. Esaminando il partecipante Adapter ad un livello di dettaglio più basso, si nota che esso è composto fondamentalmente da 4 moduli:

- XML READER: un modulo per la lettura dei dati presenti nel documento XML e la loro ristrutturazione in due strutture dati: un Dizionario dei tipi, in cui le voci sono l'insieme di tutti i dati diversi letti nel documento sorgente e sono predisposte in questa fase, mentre i valori saranno assegnati in fase di analisi lessicale; l'altra struttura, mappa Path-Data, ricostruisce lo schema del dataset XML memorizzando i dati in gruppi divisi per percorso, proprio come nel sorgente. Queste due strutture verranno illustrate in dettaglio più avanti nel capitolo.

- LEXICAL/SEMANTIC ANALYZER: un modulo contenente un analizzatore lessicale e un analizzatore semantico: il primo che assegna a tutti i dati inseriti nel Dizionario durante la lettura, uno dei 4 tipi possibili, ovvero numero, nome geografico, data e stringa; il secondo che tramite un'analisi qualitativa e quantitativa effettua una prima scrematura di tutte le voci della Mappa Path-Data, individuando quali di esse contengano i dati utili alla rappresentazione visuale e sotto quale percorso si trovino i valori numerici, le date e le categorie nel sorgente XML.

- EXTRAPOLATOR: un modulo che estrapola i dati tramite le informazioni fornite dall'analizzatore e li riassembla in pacchetti, tanti quanti sono i dati da visualizzare, ognuno contenente un valore numerico ed eventualmente una data e una o più categorie.

- DATA HARMONIZER: un modulo che si occupa di armonizzare i dati estrapolati, cioè riadattarli all'interfaccia di visualizzazione, implementata tra-

mite D3, e rifinirli eliminando eventuali dati inutili alla rappresentazione grafica.

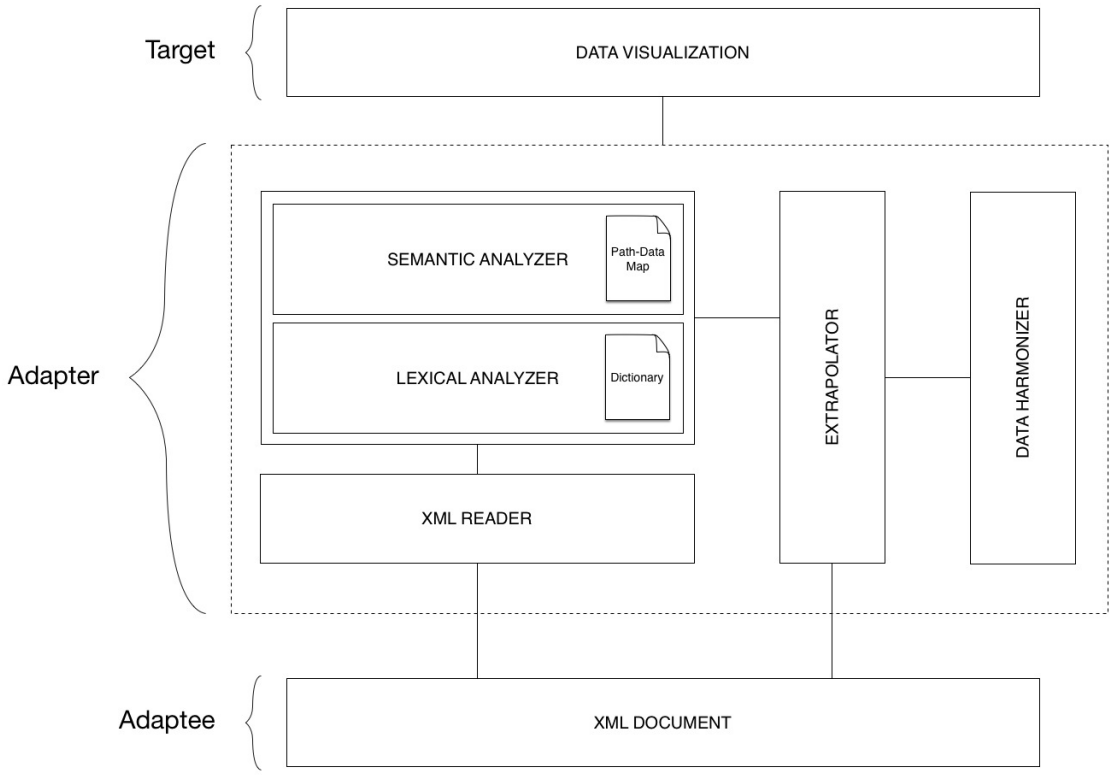

Figura 4.3: architettura

4.3 Lettura del documento XML

La prima operazione da svolgere nell'elaborazione di un generico dataset XML è la ricostruzione del suo schema: informazione sconosciuta a priori. Si effettua dunque una lettura completa del documento memorizzando ogni nodo testo e ogni attributo nelle apposite strutture dati. Va subito detto che la lettura del documento XML può rivelarsi un collo di bottiglia notevole nelle performance dell'applicazione in casi di dataset molto grandi. Una porzione di codice Javascript in esecuzione in una scheda o in una finestra di un browser, congela quest'ultima per tutta la durata del processo, di conseguenza un'esecuzione sufficientemente lunga spesso fa scattare la

cura dei browser al cosiddetto longrunning script: l'esecuzione è momentaneamente sospesa e si chiede all'utente una conferma per proseguire lo script. Nella maggior parte dei casi, questo stato rappresenta in pratica una condizione sufficiente al crash definitivo del browser. Nello sviluppo del modulo READER, illustrato in figura 4.3, si è infatti tenuto conto di questo problema cruciale nell'elaborazione dei dataset reali.

4.3.1 Uno sguardo ai Web Worker di HTML5

Come accennato all'inizio del paragrafo la lettura di sorgenti XML di dataset contenenti una grande quantità di dati è una fase molto critica in termini di performance e se non eseguita in maniera efficiente, può compromettere l'utilizzo dell'applicazione.

Una delle funzionalità più interessanti introdotte da HTML5 in supporto alle sempre più complesse applicazioni client in termini di performance è costituita dai Web Worker [18]. Prima di HTML5, sebbene i browser utilizzassero già più thread nel caricamento di una pagina web, o nell'esecuzione di più schede o finestre, i programmatori non avevano a disposizione funzioni Javascript che permettessero loro di strutturare le applicazioni in multithreading, anche perché era diffusa l'idea generale che le operazioni più complesse e lunghe dovessero essere eseguite sul server, motivo per il quale Javascript non era considerato un vero e proprio linguaggio di programmazione come invece erano i linguaggi lato server come php, python ecc.... In questo scenario l'unico modo in cui il client poteva eseguire codice asincrono e parallelo era quello di inviare al server i parametri di una determinata elaborazione tramite una chiamata Ajax per poi riceverne il risultato. I Web Worker si presentano come soluzione alla mancanza di multithreading nelle applicazioni web lato client: offrono infatti una API che permette di eseguire codice JavaScript in un thread separato che non interferisce con l'interfaccia utente dell'applicazione, per questo anche un task molto lungo non congela la pagina durante l'esecuzione. Per questo motivo i Web Worker costituiscono, all'interno del progetto DAD3, la base su cui poggia l'intero modulo di lettura dei sorgenti XML.

4.3.2 Inizializzazione e popolamento delle strutture dati

Durante la lettura del documento XML, ogni dato contenuto nei nodi testo e negli attributi deve essere memorizzato in due strutture dati diverse: la prima è un dizionario costruito con un array associativo, dove per ogni dato si predispone per la successiva analisi lessicale un'associazione vuota di cui il dato costituisce la chiave, ovvero una voce del dizionario. La seconda struttura è sempre un array associativo, ma a due dimensioni, in cui nella prima dimensione si memorizza il percorso (path) che il dato aveva nel documento XML, riportando i nodi separati dal carattere / e il nome degli attributi tra parentesi quadre in modo da avere associazioni diverse per ogni attributo dello stesso nodo; nella seconda dimensione si memorizza il dato e si aggiorna il valore dell'associazione formata dalla coppia path-dato con il numero di occorrenze trovate per quella coppia. Tale struttura dati, nominata mappa Path-Data, è alla base dell'applicazione e serve per un primo raggruppamento semantico dei dati: questi infatti saranno raggruppati per path proprio come nel XML che per natura descrive i dati con significato diverso con path diversi. Di seguito si riporta la mappa Path-Data per il sorgente XML in figura 2.2: come si può notare la mappa Path-Data contiene i dati strutturati nello schema originale del sorgente XML.

wb:data/wb:data/wb:country	
Austria	3
France	3
Germany	3
Italy	3
Spain	3
wb:data/wb:data/wb:country[id]	
AT	3
DE	3
ES	3
FR	3
IT	3
wb:data/wb:data/wb:date	
2010	5
2011	5
2012	5

wb:data/wb:data/wb:decimal	
1	15
wb:data/wb:data/wb:indicator	
GDP (current US$)	15
wb:data/wb:data/wb:indicator[id]	
NY.GDP.MKTP.CD	15
wb:data/wb:data/wb:value	
376837981578.947	1
399649131196.966	1
417656162500	1
1349350732836.2	1
1380109210526.32	1
1476881944444.44	1
2013263114238.88	1
2041954747600	1
2192357094734.72	1
2548315434210.53	1
2612878387760.35	1
2779719500000	1
3284473684210.53	1
3399588583183.34	1
3600833333333.33	1
wb:data[page]	
1	1
wb:data[pages]	
1	1
wb:data[per_page]	
50	1
wb:data[total]	
15	1

Tabella 4.1: Struttura Dati Path-Data Map dopo la lettura del sorgente in figura 2.2.

4.4 Analisi lessicale

L'analisi lessicale rappresenta la fase successiva alla lettura del documento. In questa fase, l'analizzatore lessicale scorre i dati memorizzati come voci nel dizionario e ne assegna un tipo.

4.4.1 Dati quantitativi e dati qualitativi

I dati contenuti nei dataset possono essere divisi in due categorie fondamentali:

- **dati quantitativi**: ovvero i dati che esprimono una quantità, che si presentano sotto forma di valori *numerici*
- **dati qualitativi**: ovvero i dati che esprimono una qualità, che si presentano sotto forma di valori *non numerici*.

L'unica eccezione è costituita dalle date, le quali pur essendo dati qualitativi possono apparire sotto forma di stringa, per esempio il 1 Gennaio 1990, ma anche sotto forma di valore numerico, per esempio il 1990. Quest'ultimo esempio di data, pur costituendo appunto un valore ambiguo, ha una forma accettata dall'analizzatore poiché molto diffusa nei dataset reali. Si è infatti scelto di prendere in considerazione questa forma di data pur accettando l'eventualità di un errore nell'assegnamento del tipo: in prima analisi ogni valore di questo tipo sarà segnato come data, ma soltanto in fase di analisi semantica, tramite un'analisi quantitativa dei dati simili, si deciderà se si tratta effettivamente di una data o meno, con una probabilità d'errore, a questo punto, considerevolmente ridotta rispetto alla fase iniziale. Questo passaggio sarà spiegato più dettagliatamente nel paragrafo 4.5.3.

4.4.2 Assegnamento dei tipi

L'analizzatore lessicale può assegnare a ciascun dato uno dei seguenti tipi: *valore numerico*, *data*, *stringa* e *nome geografico*. I tipi sono assegnati tramite l'uso delle seguenti funzioni che prendono posto nell'algoritmo in figura 4.4.
La funzione **parsedate()** costruisce e ritorna un oggetto nativo JavaScript di tipo *Date* a partire dalla stringa passata come parametro se riconosce quest'ultima come data, altrimenti ritorna *false*. Questa funzione fa parte della libreria globalize [12]

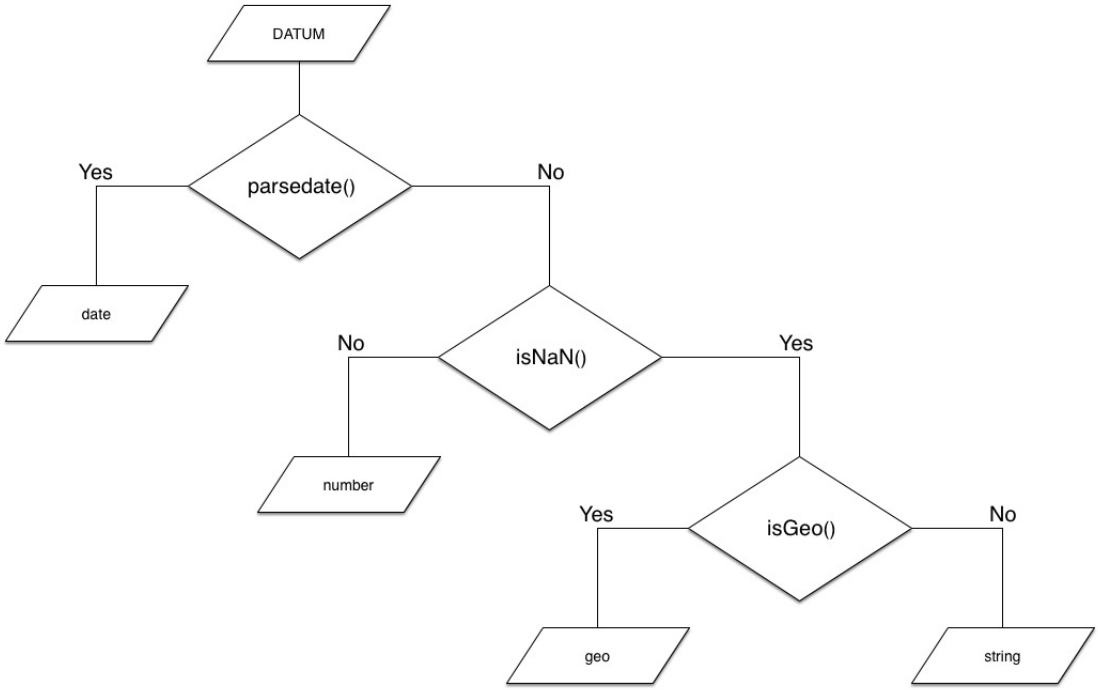

Figura 4.4: Algoritmo di assegnamento dei tipi

ed è stata opportunamente modificata, prima di includerla nel progetto, in modo tale che riconoscesse tutte le forme di data più diffuse. Come esempio di alcuni pattern accettati abbiamo:

- 12 Novembre 2013, 12/11/13, 12/11/2013, 12-11-13, 12-11-2013, ecc... ;
- Novembre 2013, 11/13, ecc... ;
- 2013.

Come spiegato nel paragrafo 4.4.1, nell'interpretazione dei tipi le date hanno la precedenza, cioè se un numero è riconosciuto come data, per esempio 1990, esso è considerato data e non numero. La funzione **isNaN()** è invece una funzione nativa di JavaScript che ritorna rispettivamente *true* o *false* a secondo che il parametro passato non sia un valore numerico oppure lo sia. Infine **isGeo()** ritorna *true* o *false* a secondo che la stringa sia o meno un nome geografico.

4.4.3 I nomi geografici

Per implementare il riconoscimento dei nomi geografici solitamente si utilizzano due metodi: si effettuano richieste per ogni dato a un web service esterno che offre questo tipo di servizio, oppure si costruisce un dizionario interno contenente tutti i possibili nomi geografici. Nello sviluppo di DAD3 il primo approccio è stato escluso sia per questioni di efficienza, considerando la mole di dati che l'applicazione deve analizzare; sia perché a parte Google, esistono pochi provider ad offrire un servizio accurato. Si è quindi costruito un database interno di nomi geografici in varie lingue e codici iso, spesso utilizzati nei sorgenti XML, che è stato ricavato tramite un merge programmatico di diverse risorse trovate online. Questa soluzione può offrire risultati mediocri all'inizio, ma ha il vantaggio di poter essere migliorata nel tempo: più grande è il database, più è alta la probabilità che l'applicazione riconosca un nome geografico.
Le stringhe intepretate come nomi geografici sono inoltre associate ad un oggetto TopoJSON [15], che contiene la rispettiva cartografia in formato JSON adeguatamente compresso.

4.5 Analisi semantica

I dataset ideali per la rappresentazione grafica sono quelli dotati di una struttura bilanciata con informazioni ridondanti, tale cioè che ogni dato contenga in sé tutte le informazioni utili che lo caratterizzano. Sebbene il datawarehouse di World Bank, come mostrato nella figura 2.2, offra dataset molto vicini alla forma ideale, nella realtà si trovano spesso dataset caratterizzati da una struttura gerarchica, divisi cioè in sottoinsiemi di dati con le stesse informazioni qualitative, in cui quindi è l'insieme e non il dato a riportare le informazioni qualitative. Inoltre i dataset reali contengono, oltre ai valori numerici che costituiscono le variabili quantitative dell'osservazione, altri numeri che rappresentano codici interni al datawarehouse, indici di ordinamento o altri generi di dato che devono essere scartati. Questo fenomeno si può notare ad esempio nel dataset di Eurostat proposto in figura 2.1: il dataset è strutturato in forma gerarchica e per ogni dato mantiene un indice di ordinamento inutile alla visualizzazione.
I vari schemi di dataset XML saranno illustrati in dettaglio nel paragrafo 4.6.

Durante l'analisi semantica, lo schema ricostruito nella mappa Path-data è analizzato dall'applicazione al fine di individuare quali raggruppamenti semantici rappresentino rispettivamente i valori numerici, le date, le categorie semplici, le categorie geografiche e quali invece rappresentino quelle informazioni inutili, quindi da scartare.
L'analisi inizia con il confronto di tutti i gruppi di dati numerici volto all'identificazione dei dati quantitativi dell'osservazione, per poi passare all'individuazione dei dati qualitativi associati.

4.5.1 Individuazione dei valori numerici

Per distinguere i valori numerici che rappresentano le variabili quantitative dell'osservazione da altri numeri inutili, l'applicazione cerca nella mappa Path-Data la voce che contenga l'insieme più corposo di valori numerici il più possibile senza ripetizioni e con un alto indice di dispersione, in modo da escludere rispettivamente numeri che si presentino con troppe ripetizioni e numeri progressivi. Nel caso si individuino più insiemi di dati numerici che soddisfano le condizioni precedentemente illustrate, ogni insieme è accettato come insieme di dati quantitativi facente parte di una tabella multidato.

4.5.2 Individuazione delle categorie

Individuati i nodi del sorgente contenenti i dati quantitativi, si passa a determinare i dati qualitativi, ovvero le categorie che caratterizzano i dati. In questa fase di analisi è di fondamentale importanza controllare che i dati qualitativi trovati includano i nodi dei dati quantitativi o ne siano inclusi: informazioni collocate al di fuori del sotto albero dei dati quantitativi non sono prese in considerazione. Ad esempio nel sorgente di Eurostat in figura 2.1, tutte le informazioni contenute nel nodo *Information* e in tutti i suoi discendenti, devono essere scartate poiché situati al di fuori dei sotto alberi che contengono i dati. I gruppi di categorie rimasti sono ulteriormente esaminati per eliminare eventuali insiemi contenenti un unico valore, trattandosi di un'informazione inutile ai fini della rappresentazione grafica. A questo punto, gli insiemi restanti costituiscono le categorie dei dati e l'ultima operazione effettuata in questa fase è una ricerca quantitativa di valori di

tipo geografico contenuti in questi insiemi: se la quantità è soddisfacente l'insieme sarà segnato come categoria geografica.

4.5.3 Individuazione delle date

A questo punto si passa all'individuazione di quei gruppi della mappa Path-Data che rappresentano le date. L'operazione è simile a quella che si effettua per l'individuazione delle categorie geografiche, l'unica differenza è che in questo caso l'analisi quantitativa è molto più restrittiva: per essere segnato come insieme di date tutti i valori dell'insieme devono essere di tipo data. Sempre considerando l'esempio di mappa Path-Data riportato in tabella 4.1, i numeri 2010, 2011 e 2012, dell'insieme wb:data/wb:data/wb:date, sono stati segnati come tipo data in fase di analisi lessicale poiché probabili date; se si fosse trovato un valore numerico non data nello stesso insieme, come ad esempio poteva essere il numero 3104, l'insieme non sarebbe stato considerato come categorie di tipo data.

4.6 Estrapolazione dei dati

Ottenute le informazioni sulla locazione dei dati qualitativi e quantitativi si procede con la loro estrapolazione dal sorgente XML. A partire dalla voce della mappa Path-Data, che rappresenta il percorso di tutti i dati quantitativi all'interno del sorgente XML, si compone un selettore in sintassi CSS che sarà passato come parametro alla funzione JQuery per la selezione dei dati dal DOM del dataset XML. Questa funzione produrrà l'array di tutti gli elementi DOM contenenti i dati quantitativi del dataset: una serie di dati, al momento, senza alcun significato poiché non ancora associati a tutti i dati qualitativi dell'osservazione. Il prossimo passo è, a questo punto, il recupero di tutti i dati qualitativi, come le date e le categorie, associati ad ogni elemento dell'array.

4.6.1 Definizione delle strutture di Dataset più frequenti

Per illustrare come DAD3 recuperi tutte le informazioni associate ai dati numerici è opportuno definire prima, attraverso due esempi, due diversi tipi di struttura che si ripetono frequentemente nei dataset:

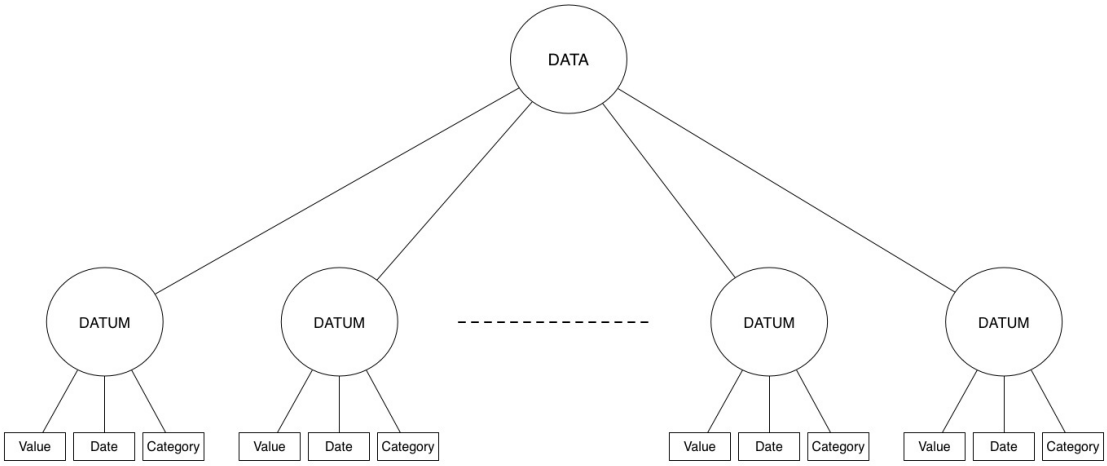

Figura 4.5: Struttura bilanciata a informazioni ridondanti: rappresenta il dataset in figura 2.2

- **Struttura bilanciata a informazioni ridondanti**: in cui tutti i dati si trovano allo stesso livello dell'albero, poiché ogni elemento contenitore del dato quantitativo contiene anche tutti i dati qualitativi.

- **Struttura gerarchica**: in cui l'albero ha un livello per ogni variabile qualitativa o quantitativa ed esistono per ogni livello tanti nodi quanti sono i diversi valori assunti dalla variabile.

La prima struttura, mostrata in figura 4.5, è quella che meglio si presta all'estrapolazione dei dati e delle informazioni associate: semplicemente partendo dall'elemento selezionato, si estrae sia il dato numerico, sia tutte le altre informazioni ad esso associate. Al contrario, in strutture come la seconda, mostrata in figura 4.6, non è facile associare i dati quantitativi ai rispettivi dati qualitativi, poiché non si conosce a priori la loro gerarchia. Per questo motivo è necessario adottare un meccanismo generale per l'estrapolazione dei dati insieme alle loro informazioni: si calcola per ogni variabile qualitativa la distanza dalla propria variabile quantitativa all'interno dell'albero e il percorso da effettuare per raggiungerla partendo da quest'ultima.

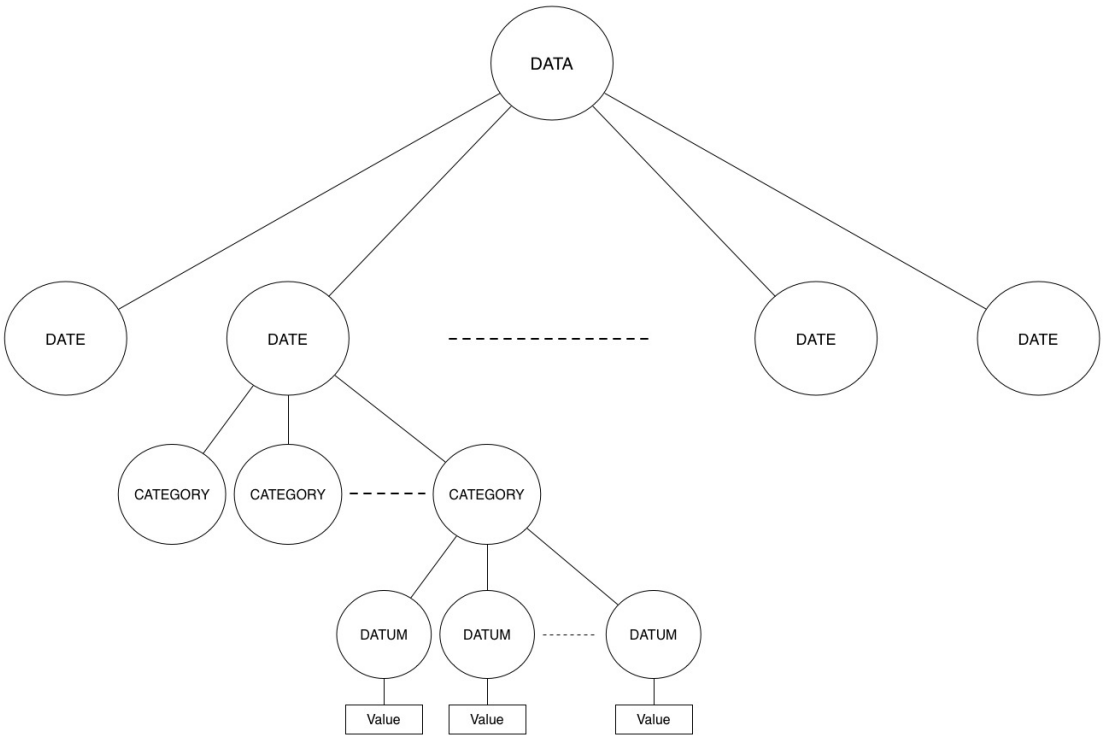

Figura 4.6: Struttura gerarchica: rappresenta il dataset in figura 2.1

4.6.2 Armonizzazione e rifinitura dei dati

Armonizzare i dati significa fondamentalmente ristrutturare i dati quantitativi insieme ai rispettivi dati qualitativi in oggetti JavaScript dalla forma nota, in modo che possano essere rappresentati graficamente dal modulo di visualizzazione. Per ogni dato quantitativo bisogna risalire l'albero alla ricerca dei dati qualitativi ad esso correlati. Si calcola quindi la distanza nell'albero XML dei dati qualitativi rispetto ai dati quantitativi. Per questo scopo il software si serve ancora una volta della struttura Path-Data, contenente i percorsi dei vari nodi, e fornisce delle direttive su come raggiungere i vari dati qualitativi a partire dal dato quantitativo in base alla loro gerarchia.
Una volta che tutti i dati e le loro informazioni correlate sono stati impacchettati in oggetti Javascript, si procede cercando di individuare se esistono informazioni ridondanti, come per esempio potrebbero essere i numeri corrispondenti all'ordine del dato in base alla data. Questa operazione può essere effettuata solo dopo che i dati sono stati estrapolati poiché occorre raggrupparli in base alle variabili qualitative, osservando i dati al variare del valore di ogni dato qualitativo: se si ottengono insiemi di dati identici per due variabili diverse si deduce che entrambe le variabili descrivono il medesimo insieme di dati e, considerate ridondanti, si sceglie quella più descrittiva eliminando l'altra. Per spiegare meglio questo meccanismo si consideri nuovamente l'esempio citato poc'anzi, in cui ci si rende conto che insieme alla data è riportata un'informazione aggiuntiva caratterizzata da un numero progressivo che varia al variare della data: tra le due viene scartato il numero progressivo poiché la data ha valore semantico.

4.7 Visualizzazione dei dati

A questo punto i dati armonizzati e ripuliti sono pronti per essere visualizzati.
Rimane da dedurre la natura dei dati e visualizzarli tramite il grafico più appropriato. In questa prima versione del progetto sono possbili tre tipi di grafici, rispettivamente a linee, a barre e a dispersione. Se i dati esprimono un andamento temporale si sceglie il grafico a linee, altrimenti un grafico a barre. Se occorre mettere in relazione due o tre dati quantitativi per lo stesso carattere si usa un grafico a dispersione.

Essendo il progetto incentrato sull'analisi esplorativa e l'armonizzazione dei dati, per la realizzazione di questo modulo si è scelto di utilizzare momentaneamente una libreria di grafici già pronti in D3 [13].

5 | Valutazioni

In questo capitolo si descrive il progetto DAD3 in termini di efficienza e di qualità dei risultati ottenuti.

5.1 Efficienza

Come accennato nel paragrafo 4.3, l'efficienza del modulo READER incide pesantemente nelle performance dell'applicazione. Infatti la lettura del sorgente XML e il conseguente caricamento in memoria sono processi che insieme impiegano circa il 35-40% del tempo d'esecuzione totale.
Ciò detto, i risultati ottenuti in fase di test sono soddisfacenti: su 20 test eseguiti analizzando un dataset [14] di circa 10000 record, su una macchina iMac con processore Intel Core i5 2,66 GHz e RAM 8 GB 1067 MHz DDR3, si ottengono tempi medi d'esecuzione del tutto accettabili.
Come mostrato nel grafico in figura 5.1 i tempi medi d'esecuzione variano in base al browser: 5.73 secondi totali per Firefox, 4.24 secondi per Chrome e 4.70 per Safari.

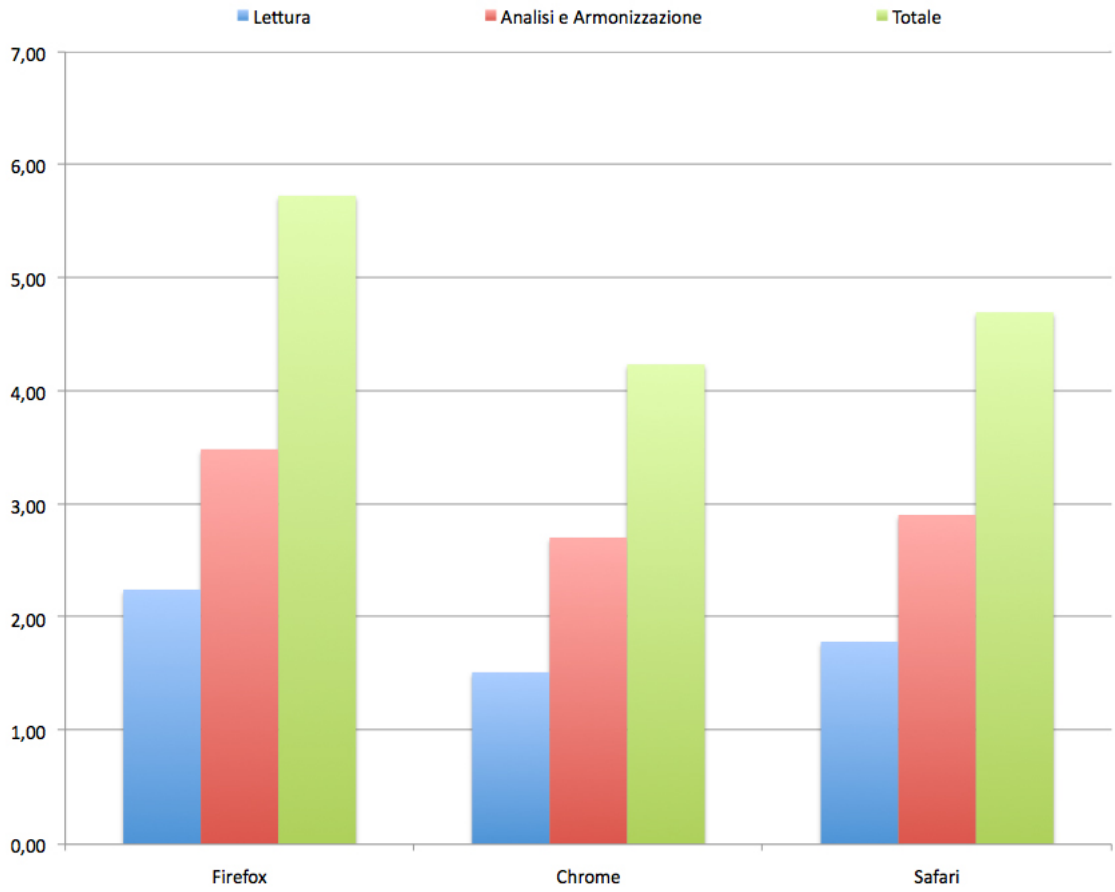

Figura 5.1: Tempi medi d'esecuzione in secondi per i vari browser.

5.2 Qualità dei risultati ottenuti

Il principio su cui si basa DAD3 è raggiungere la totale automazione del processo di armonizzazione e semplificazione dei dati: l'utente deve poter generare grafici in maniera automatica, utilizzando direttamente dataset reali e senza doverli prima rifinire. Nello sviluppo del software questo principio è stato perseguito al costo di commettere degli errori nell'analisi dei dataset. Si è osservato che in particolare esiste un tipo di struttura XML che il software non riesce a decifrare. DAD3 estrapola i dati da un sorgente XML collocati sia nei nodi testo sia negli attributi. Per questi ultimi, il software interpreta il nome dell'attributo come nome della variabile, mentre il valore dell'attributo come il valore della variabile; più attributi dello

stesso nodo sono intepretati come variabili differenti. Esistono tuttavia dataset che utilizzano due attributi per la stessa variabile specificando nei corrispondenti due valori rispettivamente il nome e il valore della variabile: in casi come questo i nomi e i valori delle variabili sono confusi nelle strutture dati interne e il programma genera risultati non validi. Ciò nonostante, i test esprimono risultati soddisfacenti nella maggior parte dei casi, percui il principio alla base del software resta valido. Inoltre XML, pur rappresentando l'unico formato accettato in questa versione, costituisce comunque il caso pessimo di struttura per un dataset; questo perché XML contiene in sé tutti i problemi riscontrabili negli altri casi: tutti gli altri formati dunque possono essere ricondotti a XML, mentre non è vero il contrario. La scelta di XML è appunto dovuto all'idea di scrivere un algoritmo generale anche se applicato di fatto a un caso specifico: la costruzione futura di altri moduli per la lettura di altri formati sarà in questo modo molto più semplice.

6 | Conclusioni

A questo punto della trattazione si può concludere che il prototipo sviluppato per il progetto di tesi presenta già in questa sua prima versione delle caratteristiche interessanti che lo contraddistinguono dagli altri software scritti allo stesso scopo. Nelle versioni successive saranno sviluppate ulteriori funzionalità che permetteranno agli utenti un controllo maggiore sui dati caricati e sulle possibili rappresentazioni grafiche.
Come già evidenziato in altre parti della tesi, esistono tantissimi strumenti che consentono di automatizzare il processo di rappresentazione grafica dei dati. Questi software però difettano se si considera un esempio d'utilizzo piuttosto comune: l'utente che vuole visualizzare i dati in forma grafica può non essere lo stesso proprietario dei dati, ma utilizzare dati pubblicati da altre organizzazioni. Le organizzazioni generano i dataset tramite i software dei loro datawarehouse che possono includere informazioni inutili; ogni organizzazione utilizza uno o più formati differenti, che non sempre coincidono con quelli utilizzati dalle altre organizzazioni. Per questi motivi chiedere all'utente di importare dati rifiniti, con una determinata struttura e uno specifico formato, può compromettere l'usabilità del software stesso. Il software che caratterizza DAD3 è scritto per superare questo genere di ostacoli e rappresenta uno strumento innovativo dal punto di vista del riuso di dataset reali per la rappresentazione grafica.

Al fine di incrementare le funzionalità e l'usabilità del progetto DAD3 è stato previsto lo sviluppo dei seguenti punti:

- Una delle funzionalità più importanti da sviluppare è l'inclusione di moduli per l'intepretazione dei sorgenti negli altri formati più diffusi: JSON, CSV e TSV.

- Il prototipo del progetto è servito a mostrare che potenzialmente è possibile sollevare l'utente dalla fase di rifinitura dei dati tramite un processo del tutto automatico, ma è comunque importare mettere a disposizione dell'utente un'interfaccia tramite la quale può seguire ed eventualmente correggere i risultati di tale processo. A questo scopo e anche per incrementare l'usabilità del prodotto, è opportuno suddividere in maniera chiara i tre step che caratterizzano l'applicazione:
  - fase di acquisizione dei dati, in cui l'utente può caricare il dataset come risorsa esterna, ma deve poter anche effettuare l'upload di un file presente sul proprio computer o scrivere il contenuto direttamente in un'apposita textarea predisposta dall'apllicazione;
  - fase di visualizzazzione dei dati acquisiti, in cui si mostrano i dati armonizzati e rifiniti in forma tabellare e viene dato all'utente la possibilità di intervenire per correggere eventuali errori;
  - fase di visualizzazione dei dati in forma grafica, in cui si sceglie il grafico migliore, ma si dà all'utente un'interfaccia più potente per poter modificare le proprietà del grafico come colori, forme ed etichette; inoltre l'utente deve poter scegliere il tipo di grafico tra tutti quelli disponibili;
- L'utente deve poter caricare più dataset e fonderli a suo piacimento;
- Occorre includere la rappresentazione grafica con mappa geografica, dato che le cartografie sono già state predisposte nel sistema.
- Sarebbe inoltre opportuno, al pari di Google Fusion Table, salvare nel sistema i grafici e i dataset creati dagli utenti, in modo che possano essere ricercati per argomento tramite un motore di ricerca interno;
- Infine un'altra funzionalità molto interessante da sviluppare sarebbe dare la possibilità all'utente di condividere il grafico creato incorporandolo in una pagina web esterna.

Bibliografia

[1] http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html.

[2] http://www.nytimes.com/interactive/2012/05/17/business/dealbook/how-the-facebook-offering-compares.html.

[3] http://epp.eurostat.ec.europa.eu/cache/ITY_FIXDST/tps00001.xml.

[4] http://api.worldbank.org/countries/it;fr;de;es;at/indicators/NY.GDP.MKTP.CD/?date=2010:2012.

[5] http://sdmx.org.

[6] http://openrefine.org.

[7] http://www.datawrapper.de.

[8] http://upload.wikimedia.org/wikipedia/commons/8/8c/Adapter_using_delegation_UML_class_diagram.svg.

[9] https://github.com/mbostock/topojson.

[10] http://www.d3js.org.

[11] http://alignedleft.com/tutorials/d3/the-power-of-data.

[12] https://github.com/jquery/globalize.

[13] http://nvd3.org.

[14] http://api.worldbank.org/countries/all/indicators/NY.GDP.MKTP.CD/?date=1980:2012&per_page=10000.

[15] Michael Bostock and Jason Davies. Code as cartography. *Cartographic Journal, The*, 50(2):129–135, 2013.

[16] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[17] Hector Gonzalez, Alon Y Halevy, Christian S Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen, and Jonathan Goldberg-Kidon. Google fusion tables: web-centered data management and collaboration. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1061–1066. ACM, 2010.

[18] I. Green. *Web Workers*. Oreilly and Associate Series. O'Reilly Media, Incorporated, 2012.

[19] Arofan Gregory and Pascal Heus. Ddi and sdmx: Complementary, not competing, standards. *Paper, Open Data Foundation (July 2007)*, 2007.

[20] Jock D Mackinlay, Pat Hanrahan, and Chris Stolte. Show me: Automatic presentation for visual analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1137–1144, 2007.

[21] Scott Murray. *Interactive Data Visualization for the Web*. O'Reilly Media, 2013.

[22] Fernanda B Viegas, Martin Wattenberg, Frank Van Ham, Jesse Kriss, and Matt McKeon. Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1121–1128, 2007.

[23] John Vlissides, R Helm, R Johnson, and E Gamma. Design patterns: Elements of reusable object-oriented software. *Reading: Addison-Wesley*, 49:120, 1995.