

Wrapper

Dott. Ing. Alessio Paolucci
alessiopaolucci@ieee.org

Topics

1. Wrapper, Introduzione.
2. Metodologie per la realizzazione.
3. L' approccio basato sulla programmazione.
4. L' approccio visuale.
5. Fondamenti di LiXto. Esempio.
6. Caso di studio: LiXto (e wrapping) nel progetto di Tesi.
7. Conclusioni. Riferimenti per approfondire.

Concetto generico

Wrapper: (Definizione tratta da Wikipedia):

The term wrapper generally refers to a type of packaging, such as a flat sheet made out of paper, cellophane, or plastic to enclose an object.

La definizione di wrapper, traslata nel contesto informatico, viene utilizzata in diversi scenari applicativi per denotare tecniche e/o metodologie anche molto diverse tra loro.

Wrapper: Tassonomia Informatica

- Programmazione (Classe wrapper)
(Es. Java offre una classe wrapper per ogni tipo primitivo. Classe Int per int, Double per double, ...)
- Trasmissione Dati
(Es. Tcp Wrapper)
- API Wrapper
(Es. Swi-Prolog offre API native in C. Le API Java sono in realtà un wrapper attorno alle API C)
- Database Wrapper
(Es. Framework per la persistenza di oggetti come Hibernate; Wrapper per interrogazioni XQuery; ...)
- **Wrapper per l'estrazione di dati da sistemi legacy e/o da sistemi esterni**
- ...

Wrapper come estrazione di dati

Dalla visione generica:

“applicazione per l'estrapolazione delle informazioni contenute in documenti normalmente reperibili su Web, su sistemi legacy e/o sistemi esterni in generale”

...allo specifico (vedi LiXto, W4F, etc...):

“Il wrapping è il processo che consiste nell' estrazione delle informazioni da documenti HTML e nella rappresentazione delle informazioni estratte in formato XML”

A cosa servono i wrapper ?

Per semplicità, si consideri l'esempio:

Si ha la necessità di realizzare un portale dedicato allo shopping che permetta all'utente di visualizzare le informazioni legate ai prodotti di proprio interesse, di confrontare prezzi e disponibilità relativamente ai principali siti di e-commerce, di confrontarne caratteristiche, etc...

Come reperire le informazioni ?

(Caratteristiche, Descrizioni, Costi e Disponibilità dei principali siti e-commerce, ...)

A cosa servono i wrapper ?



Asus F7KR

Porte USB: 5 x USB 2.0, Porte Firewall: 1 x FireWire 800, Peso: 3,5 kg

★★★★★ (Scrivi un giudizio)

+ Aggiungi alla mia lista

€988 - €1.030

[Confronta i Prezzi](#)



IBM/Lenovo 3000 N200

Porte USB: 3 x USB 2.0, Porte Firewall: 1 x FireWire 400, Sistema operativo: Microsoft Windows Vista Business

★★★★★ (Scrivi un giudizio)

+ Aggiungi alla mia lista

€502 - €1.500

[Confronta i Prezzi](#)



Acer TravelMate 7520

Porte USB: 4 x USB 2.0, Peso: 3,8 kg

★★★★★ (Scrivi un giudizio)

+ Aggiungi alla mia lista

€843 - €1.091

[Confronta i Prezzi](#)



Acer TravelMate 5520

Porte USB: 4 x USB 1.1, Porte Firewall: 1 x FireWire 800, Peso: 2,88 kg

★★★★★ (Scrivi un giudizio)

+ Aggiungi alla mia lista

€797 - €951

[Confronta i Prezzi](#)

adatto alle tue esigenze?

Tutto quello che devi sapere prima di acquistare il tuo nuovo portatile



[Guarda la guida all'acquisto »](#)

PRODOTTI PIU POPOLARI

Cosa hanno visto di piu' gli altri utenti...

-  **Asus G2PC**
€1.599 - €1.900
[Confronta i Prezzi »](#)
-  **HP Compaq Business Notebook nx6310**
€526 - €1.231
[Confronta i Prezzi »](#)
-  **Samsung Q1**
€690 - €1.112
[Confronta i Prezzi »](#)
-  **Toshiba Satellite A100**
€668 - €1.689
[Confronta i Prezzi »](#)
-  **Toshiba Tecra A8**

A cosa servono i wrapper ?

Fonti dei dati







adatto alle tue esigenze?

Tutto quello che devi sapere prima di acquistare il tuo nuovo portatile

Guarda la guida all'acquisto >

PRODOTTI PIU' POPOLARI

Cosa hanno visto di piu' gli altri utenti...

 Asus F7KR Porte USB: 5 x USB 2.0, Porte Firewall: 1 x FireWire 600, Peso: 3,5 kg ★★★★★ (Scrivi un giudizio) + Aggiungi alla mia lista	€988 - €1.030 Confronta i Prezzi
 IBM/Lenovo 3000 N200 Porte USB: 3 x USB 2.0, Porte Firewall: 1 x FireWire 400, Sistema operativo: Microsoft Windows Vista Business ★★★★★ (Scrivi un giudizio) + Aggiungi alla mia lista	€502 - €1.500 Confronta i Prezzi
 Acer TravelMate 7520 Porte USB: 4 x USB 2.0, Peso: 3,8 kg ★★★★★ (Scrivi un giudizio) + Aggiungi alla mia lista	€843 - €1.091 Confronta i Prezzi
 Acer TravelMate 5520 Porte USB: 4 x USB 1.1, Porte Firewall: 1 x FireWire 600, Peso: 2,88 kg ★★★★★ (Scrivi un giudizio) + Aggiungi alla mia lista	€797 - €951 Confronta i Prezzi

1. **HP Compaq Business Notebook nx6310**
€526 - €1.231
Confronta i Prezzi >

2. **Samsung Q1**
€690 - €1.112
Confronta i Prezzi >

3. **Toshiba Satellite A100**
€668 - €1.689
Confronta i Prezzi >

4. **Toshiba Tecra A9**



Estrazione delle informazioni

Approccio “banale”:

Estrazione manuale: Operatori estraggono dai siti di e-commerce di interesse **tutte** le informazioni e le inseriscono nel database.

Pro: Estrema semplicità di implementazione.

Contro: Costi troppo elevati.

Rispetto a strumenti automatizzati si hanno molteplici svantaggi non colmabili (Flusso dati, Lag, Sincronizzazione, ...)

... il risultato:



Automatizzazione dell'estrazione dati

E' necessario automatizzare la procedura di estrazione delle informazioni dai siti web di e-commerce.

L'operazione è concettualmente semplice, tuttavia automatizzare l'estrazione delle informazioni è nella pratica un compito tutt'altro che banale.

I portali, siti, etc... sono in genere human-readable, ma non machine-readable.



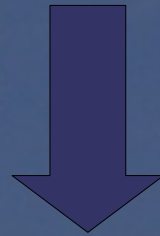
“The semantic web is dead”

E' nell' NLP e AI il futuro semantico anche del web ?

human-readable=machine-readable?

Automatizzazione dell'estrazione dati

E' quindi necessario implementare un software che estragga dalle pagine web le informazioni dei prodotti.



WRAPPER

Metodologie di implementazione

Programmazione

VS

Definizione Visuale

Programmare un wrapper

Dinamica operativa di un (semplicissimo) wrapper:

- Riceve come parametro una url
- Effettua il download della pagina (html, xhtml, text, xml, ...)
- Estrae il contenuto rilevante E' COMPLESSO!
- Restituisce i dati estratti, solitamente in formato XML.

(E' possibile anche memorizzare direttamente i dati in un database, es. database XML Nativo, ma anche inviarli ad un altro componente mediante Web Service et simila).

Programmare un wrapper

In realtà il contesto di un wrapper è molto complesso:

- Un'applicazione di retrieval dispone solitamente di un set di wrappers. (Es. Un wrapper per ogni sito di e-commerce)
- Il wrapper opportuno è invocato su ciascuna pagina individuata dal crawler.
- I wrappers fanno solitamente uso massivo di XPath ed espressioni regolari
- I wrappers devono gestire codifiche quali l'UTF-8, etc...

Tornando all'esempio

Per l'estrazione delle informazioni sui prodotti dai vari siti di e-commerce è necessario avere una lista delle url (homepage), memorizzata in un database.

Per la realizzazione è necessario un linguaggio che supporti UTF-8 e il multithreading: Java, C#, ...

Il motore dell'applicazione di retrieval ciclicamente lancia n threads di crawling che estraggono il testo delle pagine, mettendolo in cache, e i link per completare l' esplorazione.

Tornando all'esempio

Il componente di analisi esamina ciclicamente le pagine memorizzate in cache, eseguendo su ciascuna pagina il wrapper opportuno.

Le informazioni estratte dal wrapper possono essere utilizzate da altri componenti o memorizzate direttamente nella base di dati.

Utilizzando database xml nativi (eXist, Tamino, IBM DB9 Viper, ...) si possono memorizzare collections di frammenti xml per poi ottenere i full-documents mediante XQuery. Un approccio “nuovo” è quello di memorizzare informazione come clausole.

Tornando all'esempio

Frammento di html estratto da una pagina

```
<div class="information-container">      <ul class="information">
  <li class="photo">      <a
    href="javascript:uxViewLink('L2N0bC9nby9zaXRlc2VhcmNoR28.LnRzPTExOTcyODE2NTg2NzQmLnNpZz1TakhEYXc0ckNKZjN3Y0habGgyTGZZakE4
    SztJm9mZmVySWQ9NjU1MjUyMzExMzUwMWI0ODliMWEwMDJiMjdhYjg4YTZiNWZjMjk3Njk3OTkxYmQmb3J3', true);">
    
  </a>  </li>  <li class="details"><h4> <a
    href="javascript:uxViewLink('L2N0bC9nby9zaXRlc2VhcmNoR28.LnRzPTExOTcyODE2NTg2NzQmLnNpZz1TakhEYXc0ckNKZjN3Y0habGgyTGZZakE4
    SztJm9mZmVySWQ9NjU1MjUyMzExMzUwMWI0ODliMWEwMDJiMjdhYjg4YTZiNWZjMjk3Njk3OTkxYmQmb3J3', true);" class="item fn">Hp-compaq -
    Portatile dv 2268</a>      </h4>
  <div class="description">  <p>Scegliete un pc versatile che ha tutto: intelligenza, forza e bellezza. questo pc divertente e potente è il massimo quando si
    passa all'intrattenimento digitale e dà una marcia in più alla tecnologia mobile di ultima generazione per consentirvi di esse...</p>
</div>
<ul class="conditions"><li><strong>Disponibilita': <b>Disponibile</b></strong></li>
</ul>

<a href="http://www.kelkoo.it/sbs/113501/17898897.html" class="compare-prices">Confronta Prezzi &raquo;</a>

  <a href="http://shopping.kelkoo.it/ctl/do/savetolist?offerId=6552523113501it89b1a002b27ab88a6b5fc297697991bd" class="add-to-saved-list">+
  Aggiungi alla mia lista</a>
```

Tornando all'esempio

Supponiamo che il crawler memorizzi il testo delle pagine visitate nella tabella:

```
pageCache( <url>, <source>, <time_stamp> )
```

Nel campo <source> viene memorizzato il codice sorgente della pagina <url>. La codifica dei caratteri riveste un ruolo fondamentale.

<source> dovrà essere un BLOB di tipo TEXT con codifica UTF-8

Tornando all'esempio

Il componente di analisi, scomponendo l'url decide quale wrapper utilizzare sul contenuto presente in cache.

Il wrapper può estrarre informazioni:

- Se la pagina è xhtml (o xml-based) può operare come su qualsiasi documento xml con XPATH
- Se la pagina non è assimilabile ad un documento xml bisogna utilizzare un approccio più generale basato sulla manipolazione del testo. Esempio: attraverso espressioni regolari.

Espressioni regolari

Si veda direttamente l'approccio più generale; ad esempio per estrarre il titolo della pagina si può utilizzare la seguente espressione regolare:

```
<title>(.)</title>
```

Le espressioni regolari sono una sintassi attraverso la quale si possono rappresentare insiemi di stringhe. Gli insiemi caratterizzabili con espressioni regolari sono anche detti linguaggi regolari. La classe dei linguaggi regolari può essere specificata equivalentemente con tre tipi diversi di formalismi: Automi non deterministici (NFA), Automi deterministici (DFA) ed Espressioni Regolari

Questi tre formalismi hanno tutti lo stesso potere espressivo

Espressioni regolari

Per ciascun sito di e-commerce va creato il relativo wrapper. A partire dall'analisi del codice di markup delle pagine di interesse si definiscono le espressioni regolari (o le selezioni XPath se è possibile rappresentare il documento con DOM XML) e le tipologie di matching (match, match_all, ...). Esempio:

```
<h1 id="name">Acer Aspire 1524WLMi</h1>  
[...]  
<div id="price">1380</div>  
[...]
```

Espressioni regolari relative (con esempio di matching in PHP):

```
preg_match('/<h1 id="name">(.*?)<\/h1>/ui', $source, $found);  
preg_match('/<div id="price">[0-9]+<\/div>/ui', $source, $found);
```

Output

L'applicazione delle espressioni regolari (ma anche delle selezioni mediante XPath) ritorna solitamente (dipende dal linguaggio+libreria+standard) un dato complesso (o spesso Array) contenente il/i risultato/i del matching.

Partendo dai risultati ottenuti si può costruire il documento di output, ad esempio in formato XML.

In realtà è molto più frequente che l'output sia uno stream XML che viene consumato on-the-fly da altri componenti del sistema.

Esempio:

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <site>www.sitoecommerceuno.com</site>
  <date>22102007</date>
  <productslist>
    <product>
      <name>Acer Aspire 1524WLMi</name>
      <price>1380</price>
      [...]
    </product>
    <product>
      [...]
    </product>
    [...]
  </productslist>
</results>
```


Approccio Visuale

L' implementazione di un wrapper non banale è un' operazione lunga e complessa che richiede profonde conoscenze di DOM, XPath, Espressioni Regolari, etc...

In particolar modo le espressioni regolari, largamente utilizzate, sono difficili da utilizzare e la definizione corretta può richiedere tempo.

Ovviamente la realizzazione di un wrapper passa forzatamente per la programmazione, quindi necessità di personale con skill adeguato.

Esiste un approccio che permetta di automatizzare e/o di semplificare la creazione di wrapper eliminando la necessità di operare a “basso livello” con DOM, XPath, Espressioni Regolari e quant'altro ?

La risposta è affermativa: “Approccio Visuale”.

Approccio Visuale

Approccio Visuale

=

Si “disegna” il wrapper

e

si definiscono “visualmente”
le proprietà

Approccio Visuale

La metodologia di definizione visuale dei wrapper è relativamente nuova, pertanto il mercato dei prodotti di modellazione visuale è ristretto ed è sostanzialmente dominato da *LiXto*

LiXto Suite è un progetto nato da un'idea del gruppo di ricerca del Prof. Gottlob dell'Università Tecnica di Vienna.

E' un complesso di applicazioni per estrapolare e convertire informazioni contenute in documenti normalmente reperibili su Web;

E' composto da: **Lixto Visual Wrapper** e Lixto Trasformation Server.

LiXto

La creazione di un wrapper con LiXto è composta dai seguenti step:

1. Definizione della “Navigation”
2. Definizione del “Data Model”
3. Definizione dei “Patterns” del “Data Extractor”
4. Definizione dei “Filters” del “Data Extractor”
5. Definizione delle “Conditions” (Opzionale)
6. Page Crawling
7. Parameterization (Opzionale)
8. Xml Output

Un wrapper per wikipedia

Nelle slides successive verrà mostrato, con l'ausilio di screenshots, lo sviluppo visuale di un semplice wrapper che estrae i contenuti delle pagine di wikipedia.

...dopo aver analizzato gli steps seguenti

Un wrapper per wikipedia

Definizione della “Navigation”

La “Navigation” è il percorso di navigazione (di browsing) necessario a raggiungere le informazioni desiderate.

La navigazione minima consiste nella digitazione diretta dell'url.

Il concetto di “Navigation” viene introdotto in LiXto per poter raggiungere tutte quelle informazioni non raggiungibili a partire dalla sola url. Ad esempio le informazioni ottenibili solo dopo aver riempito una form o aver cliccato su pulsanti di scelta.

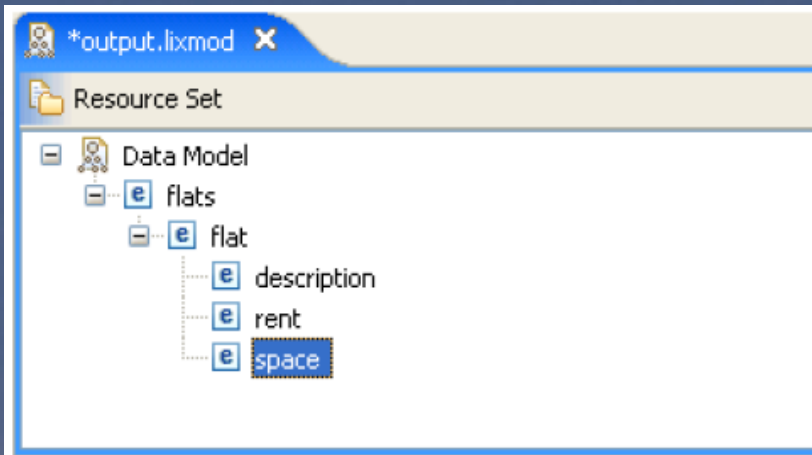
La “Navigation” viene definita semplicemente registrando tutte le azioni compiute mediante il browser embedded in LiXto sino al raggiungimento dei dati.

Un wrapper per wikipedia

Definizione del “Data Model”

La definizione del “Data Model” consiste nella costruzione del modello gerarchico dei dati di interesse.

La definizione del “Data Model” serve per varie fasi di definizione e operative del wrapper. Nell'immagine un data model tratto da un esempio incluso con Lixto.



E' importante notare che il data model rappresenta anche la struttura gerarchica dei tag che si avranno nel file xml di output.
(A meno di modifiche apportate nel Data Extractor. Raro.)

Un wrapper per wikipedia

Definizione dei “Patterns” del “Data Extractor”

La definizione del “Data Extractor” si concretizza mediante la definizione dei “Patterns” e la definizione dei “Filters” per i Patterns dati.

Semplificando si può pensare ai patterns come ai nomi da attribuire ai dati estratti.

La definizione dei patterns è solitamente banale. A meno di rare e particolari esigenze i patterns vengono definiti a partire dal Data Model che viene associato al Data Extractor.

Nel raro caso in cui il data model non rappresenti totalmente i patterns è possibile aggiungere patterns ausiliari.

Un wrapper per wikipedia

Definizione dei “Filters” del “Data Extractor”

La definizione dei “Patterns” assegna sostanzialmente un nome ai dati che si vogliono estrarre, pertanto si rende successivamente necessario caratterizzare i dati che verranno associati ai patterns.

Questo mapping è fatto visualmente, selezionando (drag & drop) i dati mediante il browser embedded in LiXto.

Nei casi più complessi (o quando la selezione visuale non permette di ottenere il risultato desiderato) è possibile operare mediante i numerosi parametri e opzioni a disposizione che permettono di definire espressioni di selezione XPath, Espressioni regolari, ...

Un wrapper per wikipedia

Definizione delle “Conditions”

La definizione dei filtri, nonostante le numerose opzioni, può in alcuni casi non essere sufficiente per soddisfare i requisiti del wrapper che si sta definendo.

Le “Conditions” permettono di definire particolari obiettivi per i criteri emersi nella fase di definizione dei filtri.

Un esempio tipico è dato dalla selezione (che è visuale) di dati rappresentati mediante tabelle. La specifica dei filtri non ha abbastanza potere espressivo per poter selezionare le righe della tabella contenenti i dati dalla/e righe di intestazione o comunque da righe da escludere. Per questo scopo è necessario far ricorso alla specifica di opportune “Conditions”.

Un wrapper per wikipedia

“Page Crawling”

Permette di definire opzioni e/o parametri legati al crawling delle pagine. Serve per poter lavorare su informazioni sparse su più pagine (es. Cataloghi, liste, etc...) e/o da prelevare da più fonti.

“Parameterization”

Permette di definire aspetti legati alla modalità operativa utilizzata da LiXto per la realizzazione del wrapper. Aspetti definibili in questo contesto sono ad esempio i Mouse Events e i Key Events.

Un wrapper per wikipedia

“Xml Output”

E' il risultato del processo di estrazione delle informazioni, ovvero dello svolgimento della sequenza delle operazioni “disegnata” negli step precedenti.

In questa fase operativa è possibile definire alcuni aspetti sul tipo di output, ad esempio se i risultati vengono distribuiti su più file xml o se racchiudere tutto in un unico documento xml.

Con questa fase terminano i passi operativi per la definizione del wrapper, che è possibile testare manualmente.

I passi successivi, (non trattati), permettono di integrare il wrapper nel Trasformation Server che usa i wrappers per reperire le informazioni, le trasforma e fornisce il risultato in output.

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The main window shows a web page with the text "Welcome to the Lixto Visual Developer." and the Lixto logo. Below the page is the DOM Source View, showing the HTML structure. On the right side, there is a "Record a Wrapper" cheat sheet with a list of actions: Introduction, New Project, New Data Model, New Wrapper, Record a navigation sequence, Create a Data Extractor, and Replay sequence. The bottom panel shows the "Actions" section with a list of actions: Browser Action, Call Action, Clean Action, Comment, Config Action, DB Loop Action, and Data Extractor. The "Info" panel on the right shows "Found instances:" with a table for "Value".

Record a Wrapper

- Introduction
 - A cheat sheet which demonstrates the creation of a simple Lixto Visual Developer project. Hit "Click to Begin" to proceed.
 - [Click to Restart](#)
- New Project
- New Data Model
- New Wrapper
- Record a navigation sequence
- Create a Data Extractor
- Replay sequence
 - Starts replaying of navigation and executes the wrapper. If you have defined patterns and filters, an XML output is generated in the corresponding output directory.
 - [Click to perform](#)
 - [Click to skip](#)

DOM Source View

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2
3   <head>
4
5
```

Actions

- Browser Action
- Call Action
- Clean Action
- Comment
- Config Action
- DB Loop Action
- Data Extractor

Info

Found instances:

Result:	0
Input:	0
Context:	0
Invisible values:	
Value	

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The main window shows the Wikipedia homepage with the URL `http://it.wikipedia.org/wiki/Pagina_principale` in the address bar. A red box highlights the recording controls, and a red arrow points to the stop button. The interface includes a Navigator, Outline, DOM Source View, and Actions panels. The DOM Source View shows the following HTML code:

```
1 <html lang="it" dir="ltr" xml:lang="it" xmlns="http://www.w3.org/1999/xhtml">
2   <head>
3
4
5
```

The Actions panel shows a list of actions: Browser Action, Call Action, and Clean Action. The Info panel on the right shows the following data:

Found instances:	
Result:	0
Input:	0
Context:	0
Invisible values:	

Red annotations on the screenshot include:

- 1. Si preme il tasto Recording. Si attiva il tasto di stop.
- 2. Si inserisce l'url

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The main window shows a recorded wrapper for the Italian Wikipedia homepage (http://it.wikipedia.org/wiki/Pagina_principale). The wrapper is recorded in the file `*web.lixvw (Recording)`. The interface includes a Navigator on the left showing the project structure, an Outline on the left showing the recorded actions, a DOM Source View at the bottom, and an Actions panel at the bottom. The recorded actions are:

- [1] it.wikipedia.org
- [2] Mouse Action
- [3] Key Action

The DOM Source View shows the following HTML structure:

```
1 <html lang="it" dir="ltr" xml:lang="it" xmlns="http://www.w3.org/1999/xhtml">
2   <head>
3
4
5
```

The Actions panel shows the following actions:

- Browser Action
- Call Action
- Clean Action

The right sidebar contains a "Record a Wrapper" panel with the following steps:

- Introduction
- New Project
- New Data Model
- New Wrapper
- Record a navigation sequence
- Create a Data Extract
- Replay sequence

The "Replay sequence" panel includes the following instructions:

- Click to perform
- Click to skip

The bottom status bar indicates: "Hold CTRL to record mouse moves."

Un wrapper per wikipedia

Lixto - LixtoWikipediaExample/web.lixvw - Lixto Visual Developer 4.8

File Edit Navigate Project Run Recording Window Help

output.lixmod output.xml pc-0.xml *web.lixvw (Recording)

http://it.wikipedia.org/wiki/Robot

Premendo il tasto Stop termina la Navigazione.

Record a Wrapper

- Introduction
 - A cheat sheet which demonstrates the creation of a simple Lixto Visual Developer project. Hit "C to Begin" to proceed.
 - Click to Restart
- New Project
- New Data Model
- New Wrapper
- Record a navigation sequence
- Create a Data Extractor
- Replay sequence
 - Starts replaying of navigation and executes the wrapper you have defined pattern and filters, an XML output generated in the corresponding output directory.
 - Click to perform
 - Click to skip

Found instances:

Result:	0
Input:	0
Context:	0

Transferring data from upload.wikimedia.org...

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The main window shows the Wikipedia page for 'Robot' (http://it.wikipedia.org/wiki/Robot). A red box highlights the recording button in the top toolbar, with the text "...ed infatti si (ri)attiva il tasto per il Recording." overlaid in red. The interface includes a Navigator on the left showing the project structure, an Outline on the left showing the page class and actions, a DOM Source View at the bottom, and an Actions panel at the bottom. The right sidebar contains a 'Record a Wrapper' panel with a list of actions: Introduction, New Project, New Data Model, New Wrapper, Record a navigation sequence, Create a Data Extract, and Replay sequence. The 'Record a navigation sequence' action is checked. The 'Replay sequence' section includes a description and buttons for 'Click to perform' and 'Click to skip'. The 'Info' panel at the bottom right shows 'Found instances: Result: 0, Input: 0, Context: 0'.

...ed infatti si (ri)attiva il tasto per il Recording.

Record a Wrapper

- Introduction
- New Project
- New Data Model
- New Wrapper
- Record a navigation sequence
- Create a Data Extract
- Replay sequence

Click to perform

Click to skip

Found instances:

Result:	0
Input:	0
Context:	0

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The title bar reads "Lixto - LixtoWikipediaExample/output.lixmod - Lixto Visual Developer 4.8". The menu bar includes File, Edit, Navigate, Project, Run, Model, Window, and Help. The toolbar contains various icons for file operations and execution. The Navigator on the left shows a project structure with folders like "Lixto-Examples", "lixtoprova", and "LixtoWikipediaExample", and files like ".project", "output.lixmod", "web.lixvw", and "wikipediawrapper". The Resource Set in the center shows a "Data Model" with a "wikipage" entity containing "maintext", "ref", and "link" elements. The Outline on the bottom left shows the "Data Model". The DOM Source View on the bottom right shows the following HTML code:

```
1 <html lang="it" dir="ltr" xml:lang="it" xmlns="http://www.w3.org/1999/xhtml">
2   <head>
3
4
5
6     <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
7   </meta>
8
9     <meta content="Robot,1495,1738,1817,1818,1860,1865,1885,1917,1920,1927" name="keywords">
10  </meta>
11
12   <link href="/favicon.ico" rel="shortcut icon">
13 </link>
14
15 <link title="Wikipedia (Italiano)" href="/wp/wikipedia_doga.php" type="application/javascript">
```

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The main window shows a browser view of the Wikipedia page for "Robot" in Italian. The browser address bar shows "http://it.wikipedia.org/wiki/Robot". The page content includes a "Wikimedia Commons" section and a "Collegamenti esterni" (External links) section with three links: "AmorphicRobotWorks (ARW)", "International Federation of Robotics", and "The Robot Hall of Fame".

The interface also features a "Navigator" pane on the left showing the project structure, including "LixtoWikipediaExample" and "wikipediawrapper". The "Outline" pane shows the "Action Sequence" for the page, with the "Data Extractor" action highlighted. The "DOM Source View" pane shows the HTML source code of the page, including the following code:

```
1 <html lang="it" dir="ltr" xml:lang="it" xmlns="http://www.w3.org/
2   <head>
3
4
5
6     <meta content="text/html; charset=utf-8" http-equiv="Cont
7   </meta>
8
9     <meta content="Robot,1495,1738,1817,1818,1860,1865,1885,1
10  </meta>
11
12   <link href="/favicon.ico" rel="shortcut icon">
13 </link>
14
15   <link title="Wikipedia (Italiano)" href="/w/opensearch_de
16 </link>
17
```

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The main window shows a browser view of the Wikipedia page for 'Robot' in Italian. The page content includes the Wikipedia logo, a donation banner for 35,893 donors, and the article text: 'Il termine **robot** (pron. *ròbot*) indica una qualsiasi macchina (di forma più o meno antropomorfa), in grado di svolgere più o meno indipendentemente un lavoro al posto dell'uomo.'

On the left, the 'Outline' pane shows a project structure with a 'wikipage' element selected, containing a 'Filter' sub-element. An arrow points from this 'Filter' element to the main browser view, indicating the current filter's target.

At the bottom, the 'Properties' pane shows the 'Selection' tool with navigation arrows and an 'Apply Selection' button. The 'Selected node' is identified as `/html[1]/body[1]/div[1]/div[1]/div[1]`.

On the right, the 'Record a Wrapper' pane shows a sequence of actions: Introduction, New Project, New Data Model, New Wrapper, Record a navigation sequence, and Create a Data Extractor. The 'Replay sequence' section is also visible, with options to 'Click to perform' and 'Click to skip'.

Un wrapper per wikipedia

The screenshot displays the Lixto Visual Developer 4.8 interface. The main window shows the Italian Wikipedia page for 'Robot' at <http://it.wikipedia.org/wiki/Robot>. A dashed black box highlights a paragraph of text: 'Il termine **robot** (pron. *ròbot*) indica una qualsiasi macchina (di forma più o meno antropomorfa), in grado di svolgere più o meno indipendentemente un lavoro al posto dell'uomo.'

The interface includes several panels:

- Navigator:** Shows the project structure with folders like 'Lixto-Examples', 'LixtoWikipediaExample', and files like 'web.lixvw'.
- Outline:** Shows the 'Page Class start' with an 'Action Sequence' containing steps like '[1] it.wikipedia.org', '[2] Mouse Action', '[3] Key Action', '[4] Vai', and '[5] Data Extractor'. A 'Filter' is applied to the 'wikipage' element.
- DOM Source View:** Shows the HTML source code with a selection on the paragraph text, displaying the path `./div[2]/p[1]`.
- Filter Panel:** Shows the 'Selection' panel with navigation arrows and an 'Apply Selection' button.
- Record a Wrapper Panel:** On the right, it shows a 'Record a Wrapper' sequence with steps: 'Introduction', 'New Project', 'New Data Model', 'New Wrapper', 'Record a navigation sequence', and 'Create a Data Extractor'. It also includes a 'Replay sequence' section.

The bottom status bar shows 'Found instances: Result: 0, Input: 1, Context: 0'.

Un wrapper per wikipedia

```
output.lixmod  output.xml  pc-0.xml  web.lixvw  output.lixmod  output.xml  *pc-0.xml X
<?xml version="1.0" encoding="UTF-8"?>
<lixto:documents xmlns:lixto="http://www.lixto.com/navigation" lixto:pageclass="start">
  <lixto:document lixto:docID="1#1197389065906" lixto:uri="http://it.wikipedia.org/wiki/Robot"/>
  <lixto:document lixto:docID="2#1197389070093"
    lixto:uri="http://it.wikipedia.org/wiki/Pagina_principale"/>
  <lixto:document lixto:docID="3#1197389070406" lixto:uri="http://it.wikipedia.org/wiki/Robot">
    <wikipage>
      <maintext>
        Il termine robot (pron. ròbot) indica una qualsiasi macchina (di forma più o meno antropomor:
        in grado di svolgere più o meno indipendentemente un lavoro al posto dell'uomo.
      </maintext>
      <ref>
        Gianmarco Veruggio, Il Mare della Robotica, Di Renzo Editore, 1999
      </ref>
      <link>
        * AmorphicRobotWorks (ARW) - progetto per lo sviluppo di performance ed installazioni robotic:
        * International Federation of Robotics.
        * The Robot Hall of Fame.
        * TrueForce - Fonte con notizie ed articoli molto approfonditi
        * Open Automaton Project
        * Portale-guida per costruttori di robot fai da te
        * OrionWiki - Una Wiki dedicata interamente al mondo della tecnologia
        * Robots.net - Sito professionale e amatoriale con foto, descrizioni di progetti ed articoli
      </link>
    </wikipage>
  </lixto:document>
</lixto:documents>
```

Questo risultato è a titolo puramente indicativo, poichè è stato sottoposto ad uno step di text filtering. Il risultato ottenibile realmente con le sole impostazioni viste (ovvero senza il passaggio di filtraggio) avrebbe contenuto stralci di codice di markup.

Caso di studio

Come caso di studio viene analizzato il progetto di tesi nel quale LiXto è utilizzato per l'estrazione delle informazioni testuali da Wikipedia.

Aspetti teorici

La tesi ha come tema l'Elaborazione del Linguaggio Naturale (NLP).

Il Natural Language Processing è un (sotto)campo dell'Intelligenza Artificiale e della Linguistica Computazionale che ha come scopo la comprensione del linguaggio naturale (umano).

Nella tesi in oggetto si esamina un approccio basato sull'analisi parallela sintattica e semantica totalmente informata (ovvero si può utilizzare, durante l'analisi, la totalità delle informazioni presenti nella knowledge base).

Si mira, sostanzialmente, a simulare, relativamente al linguaggio (e alle fondamentali interazioni con le altre funzioni superiori), le dinamiche operative della Neo Cortex, oggetto di studio delle Neuroscienze.

Aspetti teorici

“La neocorteccia è la regione del cervello dove risiedono le attività dette superiori, tipicamente umane, come apprendimento, memoria, creatività, abilità di costruzione degli utensili, comunicazione mediante il linguaggio e pianificazione di azioni. È la regione dorsale della corteccia cerebrale ed è evolutivamente la parte del cervello sviluppatasi nell'ultimo periodo di sviluppo filogenetico. Essa rappresenta circa il 90% dell'intera massa cerebrale.”

Lo studio alla base del lavoro di tesi indaga e cerca di modellare le dinamiche di interazione dei processi fondamentali per la comprensione del messaggio codificato mediante il linguaggio naturale. Il processo sintattico è sostanzialmente funzionale ai processi semantici che necessitano costantemente di interagire ed utilizzare le funzionalità tipiche della memoria, della deduzione e dell'apprendimento.

Non considerare le interazioni funzionali prima menzionate può si portare ugualmente alla creazione di strumenti e tecnologie utili in specifici contesti e/o domini applicativi, anche con brillanti risultati, ma diviene una limitazione non appena si esce fuori dal perimetro esplorato con il risultato di avere soluzioni per “problemi giocattolo”.

Progetto Mnemosine

Mnemosine è il nome del motore di ricerca semantico in sviluppo, il cui scopo è quello di essere un vero e proprio “banco di prova” per la teoria elaborata. E' quindi un ***proof-of-concepts***.

Dall'esperienza maturata nel testing di sistemi “intelligenti” emerge che solo una piccola parte di essi mantiene le promesse, in termini di funzionalità, una volta applicati al di fuori dei problemi preconfezionati sui quali è stato sviluppato.

Pertanto si è optato per testare il lavoro non su contenuti creati ad-hoc, o in ogni caso su *use cases* sviluppati appositamente, ma su testi e contenuti reali, sviluppati da terzi. Per questo si è optato per Wikipedia. (Anche per la licenza sul testo!)

Mnemosine (Alpha) in Action

[HOME](#) | [INFORMAZIONI](#) | [LOGIN](#)

Mnemosine
SEMANTIC SEARCH ENGINE

Cosa è un albero ?

Cerca

Mnemosine (Alpha) in Action

SEARCH RESULTS

[http://mnemosine.progredi.ws/wiki/Albero_\(botanica\).html](http://mnemosine.progredi.ws/wiki/Albero_(botanica).html)

Botanica (Filtra solo questo contesto)

Albero (botanica) Da Wikipedia, l'enciclopedia libera. Per albero, in botanica, si intende una pianta perenne con fusto legnoso dotato di rami secondari che si dipartono da un unico fusto centrale mostrandone una chiara dominanza apicale. Sono quindi piante i cui organi raggiungono la struttura secondaria. Da un punto di vista sistematico la morfologia ad albero s

[http://mnemosine.progredi.ws/wiki/Albero_\(informatica\).html](http://mnemosine.progredi.ws/wiki/Albero_(informatica).html)

Informatica (Filtra solo questo contesto)

Albero (informatica) Da Wikipedia, l'enciclopedia libera. Vai a: Navigazione, cerca In informatica, un albero (tree in inglese) è la struttura dati che modella il concetto di albero presente nella teoria dei grafi. Un albero si compone di due strutture fondamentali: il nodo, che in genere contiene informazioni, e l'arco che collega gerarchicame

http://mnemosine.progredi.ws/wiki/Albero_filogenetico.html

Genetica (Filtra solo questo contesto)

Albero filogenetico Da Wikipedia, l'enciclopedia libera. Vai a: Navigazione, cerca Fig. 1: Esempio di albero filogenetico Un Albero filogenetico è un diagramma che mostra le relazioni di discendenza comune di gruppi tassonomici di organismi. La rappresentazione delle relazioni in questa forma è tipica della visione evuzionistica, second

[http://mnemosine.progredi.ws/wiki/Albero_\(meccanica\).html](http://mnemosine.progredi.ws/wiki/Albero_(meccanica).html)

Meccanica (fisica) (Filtra solo questo contesto)

Albero (meccanica) Da Wikipedia, l'enciclopedia libera. Vai a: Navigazione, cerca In meccanica, l'albero è un organo di trasmissione di un moto rotatorio. Bisogna fare una divisione sostanziale: l'albero differisce dall'asse

Mnemosine, il futuro

Le funzionalità di *Mnemosine Alpha*, appena viste, sono quelle attualmente disponibili, essendo lo sviluppo allo stadio Alpha.

In ogni caso è già possibile effettuare ricerche secondo alcuni concetti quali “*Cosa è un Albero ?*”, “*Che cosa è una foglia ?*” o “*Chi è ... ?*”, e simili.

Mnemosine è già capace di distinguere i contesti, e quindi capire, ad esempio, se il termine “Albero” in un testo è riferito ad un albero binario, o ad un albero nel contesto botanico.

E' possibile effettuare le ricerche nel contesto di pertinenza. Ad esempio si può cercare “foglia” solo nel contesto “Informatico”, e quindi non dover vagare tra un numero elevato di risultati dove la maggior parte delle risorse non ha pertinenza.

Mnemosine, il futuro

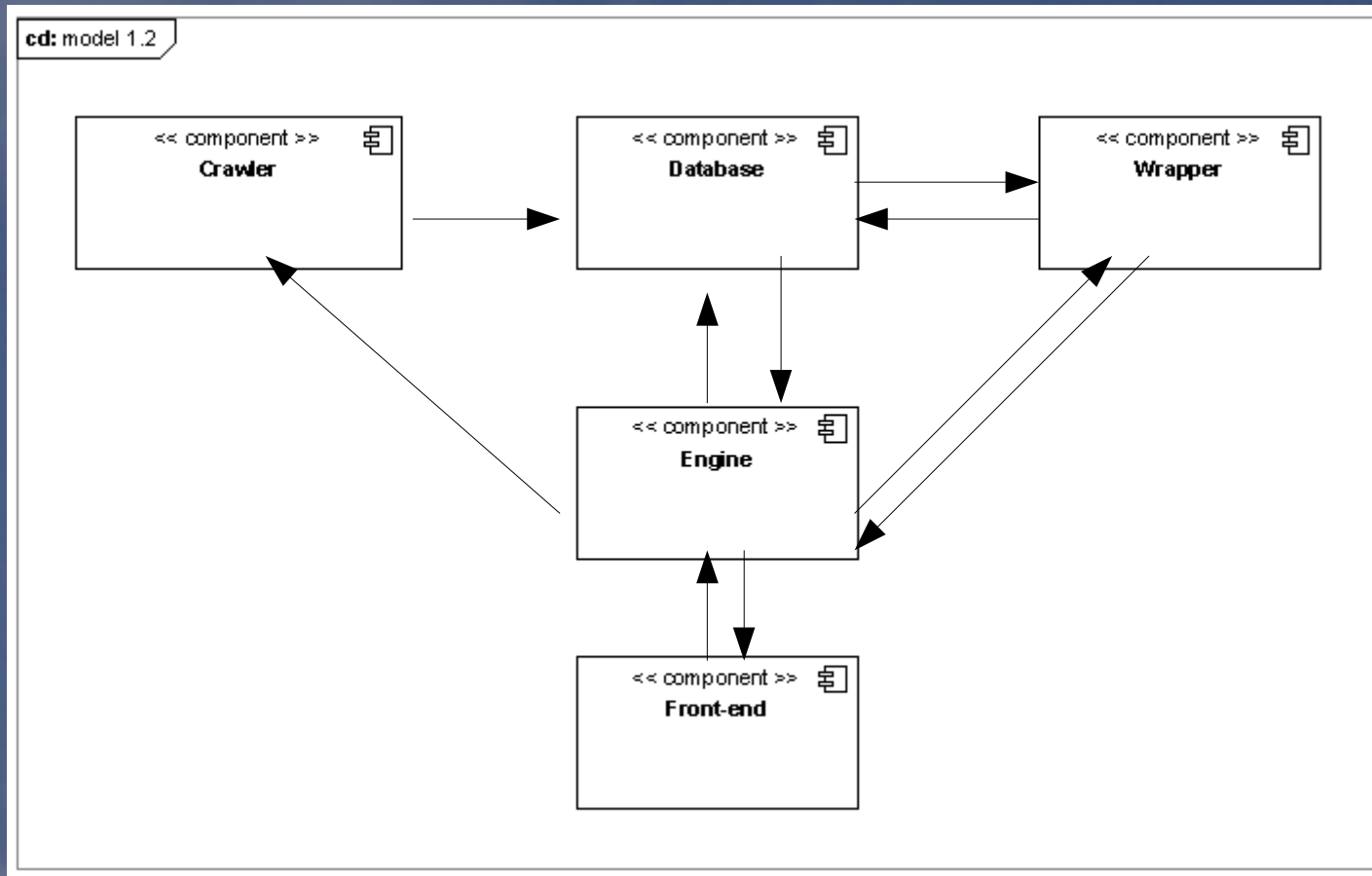
Le funzionalità in fase di implementazione sono le seguenti:

- Capacità di fornire risposte non solo intesa come indicazione delle pagine nelle quali è contenuta la risposta alla domanda posta, bensì formulare anche la risposta (o le risposte) in linguaggio naturale (e quindi generazione di testo in linguaggio naturale) basandosi su quanto appreso.
- Capacità di gestire le similitudini, concetti equivalenti, sinonimi.
- Capacità di inferenza, quindi deduzione, ragionamento.

Un banale esempio è il seguente. Dato il testo “Un albero è composto da archi e foglie”, dopo aver estratto e rappresentato la conoscenza derivante dall'analisi, deve essere possibile dedurre che “in informatica, una foglia è una componente di un albero”.

Pertanto alla domanda “Una foglia fa parte di un albero ?” deve seguire una risposta affermativa (con relative pagine di riferimento). Tutto ciò sottende la capacità di risponde in relazione a concetti dedotti e non presenti nei testi esaminati. Capacità di deduzione e generalizzazione.

Mnemosine Architettura



Mnemosine: Il Front-End

Il Front-End rappresenta il componente che si occupa dell'interazione con l'utente.

Offre i servizi tipici di una interfaccia utente via web, permette di inserire le query in linguaggio naturale e di ricevere in risposta i risultati ottenuti dal componente "Engine". La comunicazione avviene attraverso Web Service. I componenti sono stati progettati secondo il principio di massimo disaccoppiamento delle architetture SOA. Si è optato per un'architettura SOA in modo da poter distribuire i componenti e quindi i servizi su vari server (in realtà su macchine virtuali basate su Xen Hypervisor).

Il front-end è scritto in linguaggio PHP 5, ed usa un framework autosviluppato (Vortal) come base. Vortal usa XML per la descrizione di tutti i dati, XPath ed XInclude per la manipolazione di xml, XQuery per la costruzione on-the-fly degli stream xml, XSLT per le trasformazioni. Supporta molteplici terminali di output (Pc, Smartphone, Palmari, Kiosk) ed è stato pensato, e si è evoluto, per essere usato come componente in architetture SOA; pertanto offre ampio supporto per molteplici tipologie di comunicazione e Web Services. E' attualmente utilizzato anche in progetti ad ampio carico. (Esempio: <http://www.devsapiens.com>)

Mnemosine: l'Engine

Il componente Engine è stato inserito per semplificare il modello architetturale. In realtà è composto da più componenti che comunicano mediante Web Services RESTful (Architettura REST-Style). Per semplicità, in questo contesto, è possibile considerarlo, nella sua astrazione, come un componente black-box che eroga servizi.

L'engine è il controller delle fasi di crawling e wrapping. Il crawler scarica le pagine e ne memorizza il testo nella cache locale. Il wrapper viene eseguito (dall'engine) sulle pagine nella cache e ritorna come risultato uno stream xml (per ogni pagina) contenente la struttura gerarchica del testo, quindi testo -> periodi -> proposizioni.

Ad oggi il wrapper liXto-based si occupa di estrarre dal codice sorgente della pagina il contenuto di interesse (main-text). Quest'ultimo viene poi passato ad un componente di elaborazione del testo.

Mnemosine: Text Processing

Il componente di Text Processing si occupa dell':

- Eliminazione dei residui di scripting e di markup.
- Tokenizzazione, quindi della divisione del flusso di caratteri in parole.
- Eliminazione delle strutture superflue.
- Individuazione delle frasi a partire dalle parole.
- “Riparazioni” della sintassi per proposizione malformate.

Lo stream xml ottenuto viene poi consumato dal componente proprio di NLP.

Mnemosine: NLP

Il componente di NLP è sostanzialmente rappresentato da un interprete Prolog sviluppato in Java, denominato “PervasiveLogic Engine” in quanto realizzato per uso embedded in architetture J2EE, J2ME, .NET. (Per il testing manuale ed il debug è usato Swi-Prolog).

Il componente carica la base di conoscenza comune (nomi, verbi, ..., concetti comuni primitivi, regole comuni primitive, ...); la conoscenza estratta dai testi esaminati; i moduli per l'analisi del linguaggio basati su grammatiche DCG estese; etc...;

Riceve le query, già opportunamente trasformate nella rappresentazione logica da altre istanze del medesimo componente di NLP, effettua le operazioni necessarie di inferenza, ragionamento e deduzione, quindi restituisce i risultati che l'engine avrà cura di inviare al front-end.

Rappresentazione della conoscenza

Esempio di rappresentazione della conoscenza estratta:

TESTO: *“La psicologia cognitiva è una scienza che studia la mente.”*

RAPPRESENTAZIONE:

```
concept('http://it.wikipedia.org/wiki/Psicologia_cognitiva', [psicologia, neuroscienze],
  root( element(main, _, _, prdnomin( sog(art('la'), nome('psicologia'),
    agg('cognitiva')), cop('è', 'essere'),
    prdnome(art('una'), nome('scienza')))),
    element(rel, 'che', fare(sog(PREV_ELE), vrb('studia', 'studiare'),
      c_ogg( art('la'), nome('mente') ) ) ) ).
```

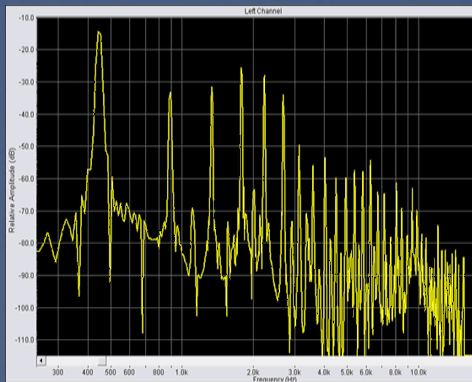
DOMANDA: *“Che cosa è la psicologia cognitiva ?”*

```
prdnomin( sog(nome('psicologia'), agg('cognitiva')), cop('è', 'essere'), prdnome( X,Y )).
```

Natural Language Processing

Come detto Mnemosine è il proof-of-concepts di una teoria. Il contesto applicativo dei motori di ricerca semantici è stato scelto per dimostrare la qualità dei risultati ottenuti peraltro in un mercato in forte espansione e particolarmente innovativo.

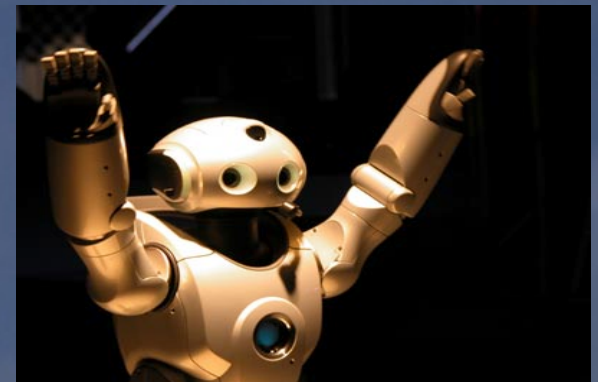
Gli aspetti teorici e il core dell' implementazione delle tecnologie di NLP possono essere applicati nei più disparati contesti, ad esempio:



Speech-To-Text
NLP
Text-To-Speech



Chatterbots



Robotics

Conclusione

In conclusione,
quale metodologia utilizzare
per lo sviluppo di
wrapper

?

Riferimenti e Approfondimenti

Web:

Lixto Software GmbH – Software LiXto

<http://www.lixto.com>

Espressioni regolari (Tutorials, articoli, software)

<http://www.regular-expressions.info/>

Jakarta Oro - Text-processing Java classes that provide Perl5 compatible regular expressions

<http://jakarta.apache.org/oro/>

IEEE Intelligent System - The #1 Magazine in Artificial Intelligence!

<http://www.computer.org/portal/site/intelligent>

Riferimenti e Approfondimenti

XPath

<http://www.w3.org/TR/xpath>

XQuery

<http://www.w3.org/TR/xquery/>

LingPipe - Suite of Java libraries for the linguistic analysis of human language.

<http://www.alias-i.com/lingpipe/>

The Berkeley Natural Language Processing Group

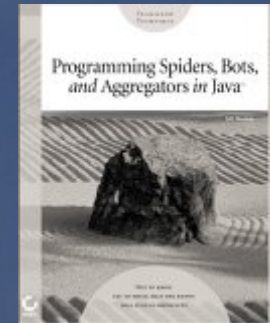
<http://nlp.cs.berkeley.edu/Main.html>

Riferimenti e Approfondimenti

Books:

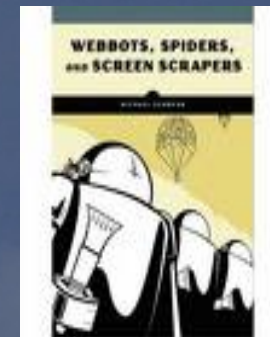
Programming Spiders, Bots, and Aggregators in Java

by Jeff Heaton



Webbots, Spiders, and Screen Scrapers: A Guide to Developing Internet Agents with PHP/CURL

by Michael Schrenk

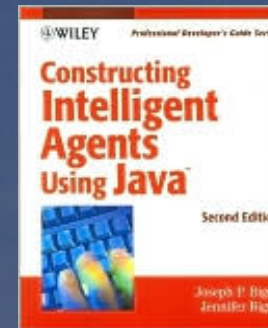


Riferimenti e Approfondimenti

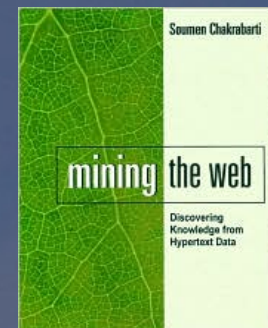
HTTP Programming Recipes
for C# Bots
by Jeff Heaton



Constructing Intelligent Agents Using
Java: Professional Developer's Guide,
2nd Edition
by Joseph P. Bigus, Jennifer Bigus, Joe Bigus



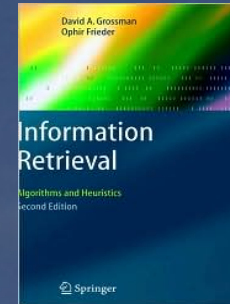
Mining the Web: Discovering
Knowledge from Hypertext Data
by Soumen Chakrabarti



Riferimenti e Approfondimenti

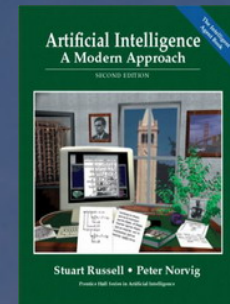
Information Retrieval: Algorithms and Heuristics. (2nd Edition)

by David A. Grossman, Ophir Frieder



Artificial Intelligence: A Modern Approach (2nd Edition)

by Stuart J. Russell, Peter Norvig



Intelligenza Artificiale

di Nils J. Nilsson

