

Il linguaggio Chat Markup Language

Progetto e implementazione di un sistema per la realizzazione di
agenti conversazionali

Alan Zucconi

Università degli Studi di Pisa
Facoltà di Scienze Matematiche, Fisiche e Naturali

24 Giugno 2011



Gli agenti conversazionali

Ogni giorno milioni di persone usano Internet per chattare con amici, parenti e colleghi.

Ma cosa succederebbe se dall'altra parte non ci fosse un interlocutore umano, ma un programma?

Definizione

Un **chatbot** è un agente software in grado di simulare una conversazione intelligente.



L'Effetto ELIZA

I chatterbot attuali mancano di meccanismi validi per poter avere una reale *comprensione semantica* o effettuare una *deduzione logica* e sono quindi impossibilitati a comprendere il significato delle frasi.

Sono necessari strumenti e tecniche alternative per illudere l'utente che il chatterbot sia dotato di un'intelligenza superiore a quella realmente posseduta.

Questa *dissonanza cognitiva* tra ciò che l'utente si aspetta da un calcolatore e il risultato che invece ottiene è detto **effetto ELIZA**.

Creare un buon chatterbot è un problema che riguarda più la *Psicologia* che l'Informatica.



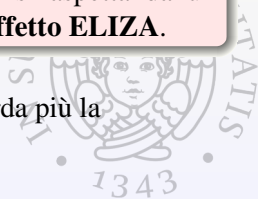
L'Effetto ELIZA

I chatterbot attuali mancano di meccanismi validi per poter avere una reale *comprensione semantica* o effettuare una *deduzione logica* e sono quindi impossibilitati a comprendere il significato delle frasi.

Sono necessari strumenti e tecniche alternative per illudere l'utente che il chatterbot sia dotato di un'intelligenza superiore a quella realmente posseduta.

Questa *dissonanza cognitiva* tra ciò che l'utente si aspetta da un calcolatore e il risultato che invece ottiene è detto **effetto ELIZA**.

Creare un buon chatterbot è un problema che riguarda più la *Psicologia* che l'Informatica.



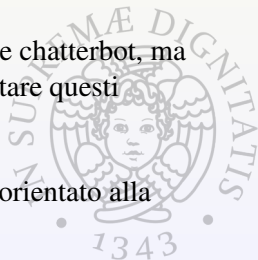
Problemi dei chatterbot

I problemi più comuni nei chatterbot attualmente presenti sul mercato sono:

- capacità conversazionale ridotta (*effetto Doretta*),
- difficoltà nel contestualizzare la conversazione,
- profondità di conversazione limitata,
- assenza di emozioni e memoria.

Esistono numerosi programmi e linguaggi per creare chatterbot, ma ben pochi offrono gli strumenti necessari per affrontare questi problemi nella loro interezza.

Nasce da questa esigenza di un nuovo linguaggio orientato alla creazione di chatterbot umanodi.



Linguaggi concorrenti

La base di partenza è stata lo studio dei linguaggi preesistenti.

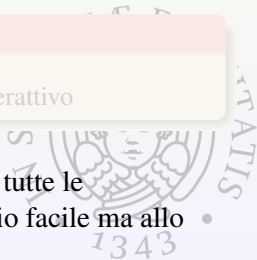
AIML

- Sintassi XML
- Turing completo
- Troppo verboso
- Complesso da usare

Polygen

- Grammatiche generative
- Non interattivo

L'obiettivo è quello di fondere in un unico prodotto tutte le caratteristiche più utili per poter creare un linguaggio facile ma allo stesso tempo potente.



Linguaggi concorrenti

La base di partenza è stata lo studio dei linguaggi preesistenti.

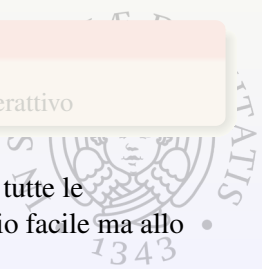
AIML

- Sintassi XML
- Turing completo
- Troppo verboso
- Complesso da usare

Polygen

- Grammatiche generative
- Non interattivo

L'obiettivo è quello di fondere in un unico prodotto tutte le caratteristiche più utili per poter creare un linguaggio facile ma allo stesso tempo potente.



Linguaggi concorrenti

La base di partenza è stata lo studio dei linguaggi preesistenti.

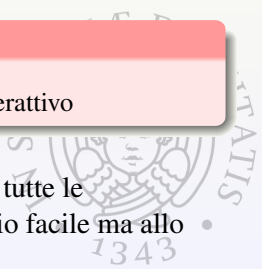
AIML

- Sintassi XML
- Turing completo
- Troppo verboso
- Complesso da usare

Polygen

- Grammatiche generative
- Non interattivo

L'obiettivo è quello di fondere in un unico prodotto tutte le caratteristiche più utili per poter creare un linguaggio facile ma allo stesso tempo potente.



Il Chat Markup Language

Il linguaggio realizzato si chiama **CML** ed è basato su:

- sintassi derivata da XML,
- paradigma dichiarativo,
- programmazione orientata alle *richieste*.

Offre nativamente alcuni strumenti tra cui:

- grammatiche generative,
- emozioni e memoria,
- contestualizzazione,
- costrutti condizionali,
- supporto JavaScript integrato,
- integrazione con plugin Java,
- ...



Esempio Regole e risposte

Questo codice memorizza il nome dell'utente:

```
<!-- Rule -->
<RULE regexp = `my name is (\w{4,})`>
  <!-- If user's name is unknown -->
  <REQUIRED memory = `name`      type = `NOT_AVAILABLE`>
    <!-- Saves the user's name -->
    <UPDATE memory = `name`      value = `$1$`>
    <!-- List of answers -->
    <ANSWER text = `Hey #name#!  I like your name!`  />
    <ANSWER text = `Hey #name#!  How are you?`  />
  </REQUIRED>

  <!-- Else -->
  <ANSWER text = `I know your name, #name#!`
    priority = `LOW`  />
</RULE>
```

Esempio Regole e risposte

Questo codice memorizza il nome dell'utente:

```
<!-- Rule -->
<RULE regexp = 'my name is (\w{4,})' >
  <!-- If user's name is unknown -->
  <REQUIRED memory = 'name' type = 'NOT_AVAILABLE' >
    <!-- Saves the user's name -->
    <UPDATE memory = 'name' value = '$1$' >
    <!-- List of answers -->
    <ANSWER text = 'Hey #name#! I like your name!' />
    <ANSWER text = 'Hey #name#! How are you?' />
  </REQUIRED>

  <!-- Else -->
  <ANSWER text = 'I know your name, #name#!'
    priority = 'LOW' />
</RULE>
```

Esempio Regole e risposte

Questo codice memorizza il nome dell'utente:

```
<!-- Rule -->
<RULE regexp = `my name is (\w{4,})`>
  <!-- If user's name is unknown -->
  <REQUIRED memory = `name`      type = `NOT_AVAILABLE`>
    <!-- Saves the user's name -->
    <UPDATE memory = `name`      value = `$1$`>
    <!-- List of answers -->
    <ANSWER text = `Hey #name#!  I like your name!`  />
    <ANSWER text = `Hey #name#!  How are you?`  />
  </REQUIRED>

  <!-- Else -->
  <ANSWER text = `I know your name, #name#!`
    priority = `LOW`  />
</RULE>
```

Esempio Regole e risposte

Questo codice memorizza il nome dell'utente:

```
<!-- Rule -->
<RULE regexp = 'my name is (\w{4,})' >
  <!-- If user's name is unknown -->
  <REQUIRED memory = 'name'      type = 'NOT_AVAILABLE' >
    <!-- Saves the user's name -->
    <UPDATE memory = 'name'      value = '$1$' >
    <!-- List of answers -->
    <ANSWER text = 'Hey #name#!  I like your name!' />
    <ANSWER text = 'Hey #name#!  How are you?' />
  </REQUIRED>

  <!-- Else -->
  <ANSWER text = 'I know your name, #name#!'
            priority = 'LOW' />
</RULE>
```

Esempio Regole e risposte

Questo codice memorizza il nome dell'utente:

```
<!-- Rule -->
<RULE regexp = 'my name is (\w{4,})' >
  <!-- If user's name is unknown -->
  <REQUIRED memory = 'name'      type = 'NOT_AVAILABLE' >
    <!-- Saves the user's name -->
    <UPDATE memory = 'name'      value = '$1$' >
    <!-- List of answers -->
    <ANSWER text = 'Hey #name#!  I like your name!'  />
    <ANSWER text = 'Hey #name#!  How are you?'  />
  </REQUIRED>

  <!-- Else -->
  <ANSWER text = 'I know your name, #name#!'
    priority = 'LOW'  />
</RULE>
```

Esempio Regole e risposte

Questo codice memorizza il nome dell'utente:

```
<!-- Rule -->
<RULE regexp = `my name is (\w{4,})`>
  <!-- If user's name is unknown -->
  <REQUIRED memory = `name`      type = `NOT_AVAILABLE`>
    <!-- Saves the user's name -->
    <UPDATE memory = `name`      value = `$1$`>
    <!-- List of answers -->
    <ANSWER text = `Hey #name#!  I like your name!`  />
    <ANSWER text = `Hey #name#!  How are you?`  />
  </REQUIRED>

  <!-- Else -->
  <ANSWER text = `I know your name, #name#!`
    priority = `LOW`  />
</RULE>
```

Esempio Regole e risposte

Questo codice memorizza il nome dell'utente:

```

<!-- Rule -->
<RULE regexp = `my name is (\w{4,})`>
  <!-- If user's name is unknown -->
  <REQUIRED memory = `name`      type = `NOT_AVAILABLE`>
    <!-- Saves the user's name -->
    <UPDATE memory = `name`      value = `$1$`>
    <!-- List of answers -->
    <ANSWER text = `Hey #name#!
                    {{I like your name!|How are you?}}` />
  </REQUIRED>

  <!-- Else -->
  <ANSWER text = `I know your name, #name#!`
          priority = `LOW` />
</RULE>

```

Esempio Categorie ed etichette

```
<ANSWER text = ``La %PENNA% è sul %TAVOLO(M)%.`` />
```

```
<CATEGORY name = ``PENNA``>
```

```
  <ANSWER text = ``penna`` />
```

```
  <ANSWER text = ``matita`` />
```

```
</CATEGORY>
```

```
<CATEGORY name = ``TAVOLO``>
```

```
  <!-- Nomi maschili -->
```

```
  <ANSWER text = ``tavolo``          labels = ``M`` />
```

```
  <ANSWER text = ``comodino``        labels = ``M`` />
```

```
  <!-- Nomi femminili -->
```

```
  <ANSWER text = ``scrivania``      labels = ``F`` />
```

```
</CATEGORY>
```

La %PENNA% è sul %TAVOLO (M) %.



Esempio Categorie ed etichette

```
<ANSWER text = ``La %PENNA% è sul %TAVOLO(M)%.`` />
```

```
<CATEGORY name = ``PENNA``>
```

```
  <ANSWER text = ``penna`` />
```

```
  <ANSWER text = ``matita`` />
```

```
</CATEGORY>
```

```
<CATEGORY name = ``TAVOLO``>
```

```
  <!-- Nomi maschili -->
```

```
  <ANSWER text = ``tavolo``          labels = ``M`` />
```

```
  <ANSWER text = ``comodino``       labels = ``M`` />
```

```
  <!-- Nomi femminili -->
```

```
  <ANSWER text = ``scrivania``     labels = ``F`` />
```

```
</CATEGORY>
```

La **%PENNA%** è sul **%TAVOLO (M) %**.

Esempio Categorie ed etichette

```
<ANSWER text = ``La %PENNA% è sul %TAVOLO(M)%.`` />
```

```
<CATEGORY name = ``PENNA``>
```

```
  <ANSWER text = ``penna`` />
```

```
  <ANSWER text = ``matita`` />
```

```
</CATEGORY>
```

```
<CATEGORY name = ``TAVOLO``>
```

```
  <!-- Nomi maschili -->
```

```
  <ANSWER text = ``tavolo``      labels = ``M`` />
```

```
  <ANSWER text = ``comodino``    labels = ``M`` />
```

```
  <!-- Nomi femminili -->
```

```
  <ANSWER text = ``scrivania``   labels = ``F`` />
```

```
</CATEGORY>
```

La **matita** è sul **%TAVOLO(M)%**.

Esempio Categorie ed etichette

```
<ANSWER text = ``La %PENNA% è sul %TAVOLO(M)%.``` />

<CATEGORY name = ``PENNA``>
  <ANSWER text = ``penna`` />
  <ANSWER text = ``matita`` />
</CATEGORY>

<CATEGORY name = ``TAVOLO``>
  <!-- Nomi maschili -->
  <ANSWER text = ``tavolo`` labels = ``M`` />
  <ANSWER text = ``comodino`` labels = ``M`` />
  <!-- Nomi femminili -->
  <ANSWER text = ``scrivania`` labels = ``F`` />
</CATEGORY>
```

La **matita** è sul **%TAVOLO(M)%**.

Esempio Categorie ed etichette

```
<ANSWER text = ``La %PENNA% è sul %TAVOLO(M)%.`` />

<CATEGORY name = ``PENNA``>
  <ANSWER text = ``penna`` />
  <ANSWER text = ``matita`` />
</CATEGORY>

<CATEGORY name = ``TAVOLO``>
  <!-- Nomi maschili -->
  <ANSWER text = ``tavolo`` labels = ``M`` />
  <ANSWER text = ``comodino`` labels = ``M`` />
  <!-- Nomi femminili -->
  <ANSWER text = ``scrivania`` labels = ``F`` />
</CATEGORY>
```

La **matita** è sul **%TAVOLO(M)%**.

Esempio Categorie ed etichette

```
<ANSWER text = ``La %PENNA% è sul %TAVOLO(M)%.`` />

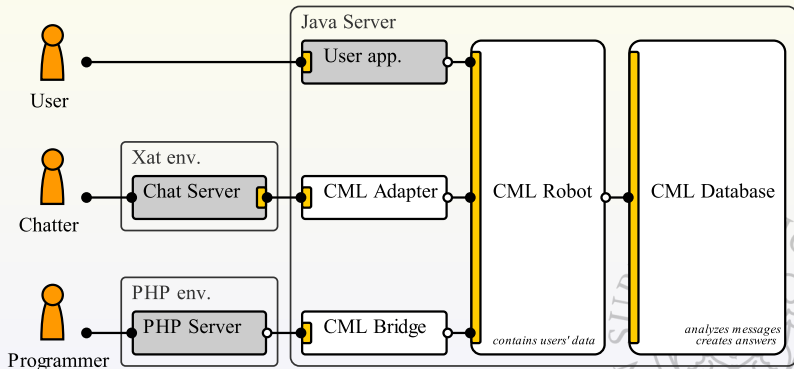
<CATEGORY name = ``PENNA''>
  <ANSWER text = ``penna'' />
  <ANSWER text = ``matita'' />
</CATEGORY>

<CATEGORY name = ``TAVOLO''>
  <!-- Nomi maschili -->
  <ANSWER text = ``tavolo'' labels = ``M'' />
  <ANSWER text = ``comodino'' labels = ``M'' />
  <!-- Nomi femminili -->
  <ANSWER text = ``scrivania'' labels = ``F'' />
</CATEGORY>
```

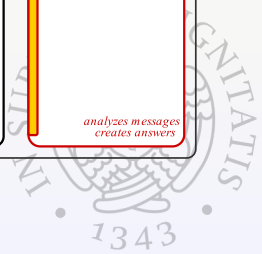
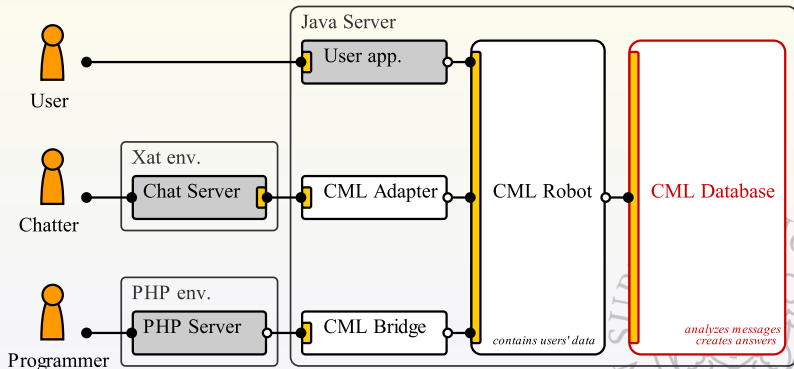
La **matita** è sul **tavolo**.



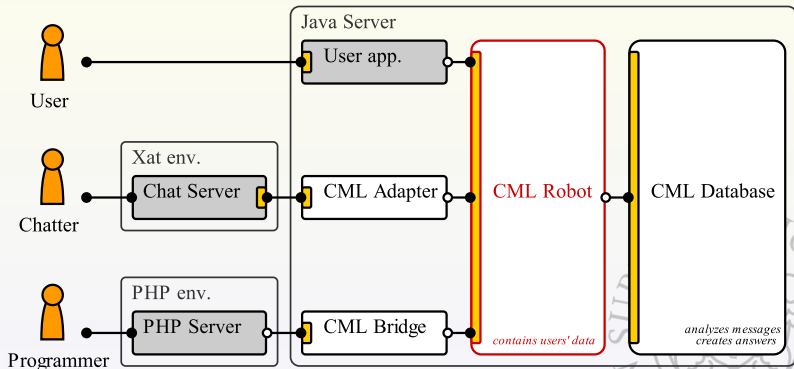
Architettura dell'interprete



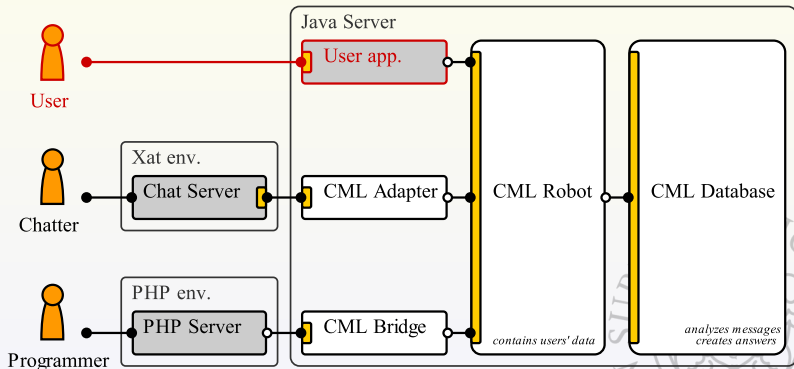
Architettura dell'interprete



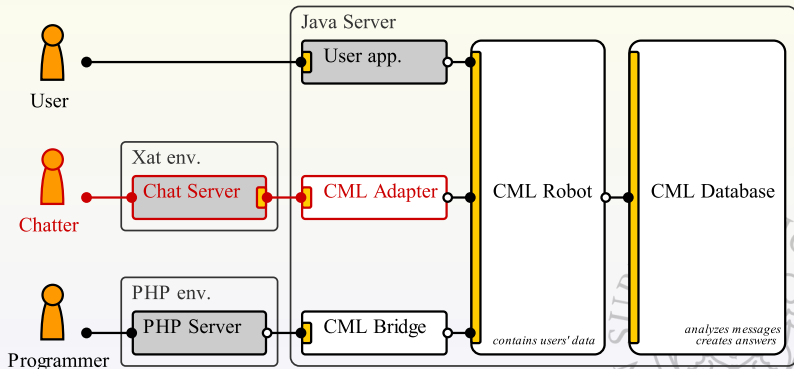
Architettura dell'interprete



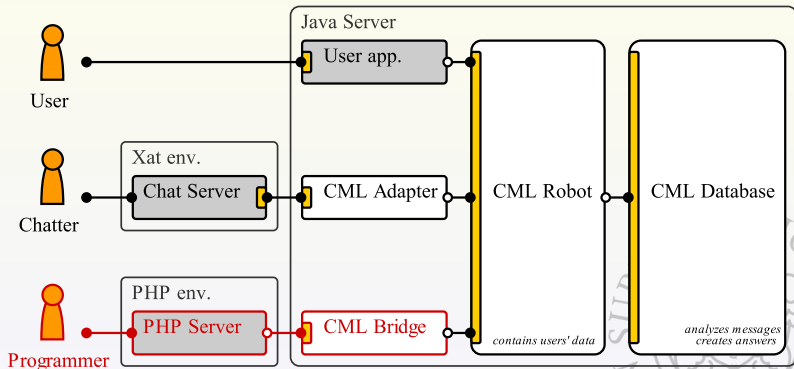
Architettura dell'interprete



Architettura dell'interprete



Architettura dell'interprete



Obiettivi

Oltre al lavoro già esposto, durante il tirocinio sono stati conseguenti anche i seguenti risultati:

- definire la sintassi del linguaggio,
 - Forma Estesa di Backus-Naur (*EBNF*)
- definire la semantica formale del nucleo del linguaggio,
 - linguaggio del primo ordine (*FOL*)
- eseguire un'accurata analisi prestazionale dell'interprete.
 - identificazione dei colli di bottiglia



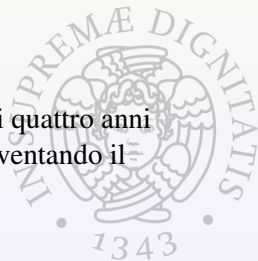
Doriana82

Doriana Ottantadue è un chatterbot, parodia della famosa *Doretta82* creata da Microsoft per sponsorizzare il motore di ricerca Live.

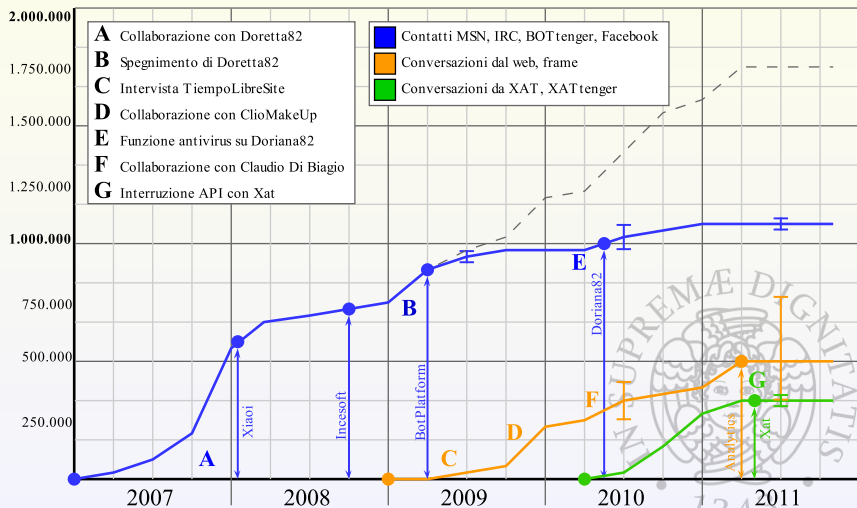


Doriana★^{beta}

Grazie alla sua personalità forte e decisa, nel giro di quattro anni Doriana ha raggiunto quasi due milioni di utenti, diventando il chatterbot più famoso d'Italia.



Utenza



XATtenger

XATtenger è un servizio online per la creazione di chatterbot moderatori, compatibili con le chat messe a disposizione dal sito *Xat*.

Visite:



Fonti: Analytics, Xat, XATtenger.

Il portale ha avuto rilevanza a livello internazionale, contando centocinquantamila iscrizioni per un totale di quasi trecentomila chatterbot creati.



Conclusione

Futuri sviluppi del CML:

- sintassi diversa da XML,
- programmazione strutturata,
- supporto per i costrutti *if-then-else*, *for*, ...
- ambiente di sviluppo integrato,
- ...

*«Mi dispiace, le mie risposte sono limitate.
Devi farmi le domande giuste.»*

Ologramma di Alfred Lanning, **Io, Robot**



Conclusione

Futuri sviluppi del CML:

- sintassi diversa da XML,
- programmazione strutturata,
- supporto per i costrutti *if-then-else*, *for*, ...
- ambiente di sviluppo integrato,
- ...

*«Mi dispiace, le mie risposte sono limitate.
Devi farmi le domande giuste.»*

Ologramma di Alfred Lanning, **Io, Robot**

Crediti

Tesi di Laurea

- **Titolo** *Il linguaggio CML: progetto ed implementazione di un sistema per la realizzazione di agenti conversazionali.*
- **Autore** Alan Zucconi
- **Relatrice** Maria Simi
- **Anno Accademico** 2010 - 2011

Contatti

- **Web** www.alanzucconi.com
- **E-mail** alan.zucconi@imperial.ac.uk

