Link <http://www.aiml.foundation/doc.html>

Il mercato dei chatbot è cresciuto e si è evoluto considerevolmente da quando AIML 2.0 è stato rilasciato quattro anni fa e questo aggiornamento delle specifiche introduce alcune nuove funzionalità chiave progettate per affrontare il panorama in evoluzione.

**1. Introduzione**

Questo documento è una bozza di specifica per un nuovo standard AIML (Artificial Intelligence Markup Language), versione 2.1. AIML è un linguaggio XML per specificare il contenuto di chatbot. Un interprete AIML è un programma in grado di caricare ed eseguire il bot e di fornire le risposte del bot in una sessione di chat con un utente umano, chiamato client. Questo documento descrive in dettaglio la sintassi e la semantica di AIML, oltre alle funzionalità chiave che un interprete AIML dovrebbe supportare.   
L'obiettivo di progettazione principale del linguaggio AIML originale era la semplicità. AIML è motivato da due osservazioni:

1. La creazione di un chatbot con capacità di conversazione richiede la scrittura di una quantità significativa di contenuto, sotto forma di risposte alla conversazione.

\*

1. Scrivere chatbot richiede un mix di design conversazionale (o letterario) e capacità di programmazione.

AIML è stato progettato per la prima volta alla fine degli anni '90, durante l'esplosione del World Wide Web. Mentre alla fine il web ha perso la sua semplicità originale, nel 1994 è stato possibile creare un sito Web con una conoscenza rudimentale di pochi tag HTML.   
AIML è stato originariamente progettato per essere ugualmente semplice e per essere chatbot che HTML è sul web, vale a dire, chiunque conosca abbastanza HTML per creare un sito Web, può imparare abbastanza AIML per scrivere un chatbot.   
Parallelamente negli anni '90, XML è emerso come standard che è ancora ampiamente accettato oggi nonostante molti formati concorrenti praticabili. La rappresentazione basata su tag XML è facile da comprendere senza una sofisticata conoscenza dell'informatica. Gli autori di AIML hanno trovato utili molti strumenti XML, come DTD, correttori di sintassi ed editor durante la creazione di bot.   
Pertanto, mentre AIML rimane agganciato al carro XML, non dipende dalla sintassi XML. Esiste una rappresentazione più profonda dei dati che rappresentiamo nei file XML. Finché la rappresentazione può catturare la struttura di base di un percorso di modello (modello di input, modello e modello di argomento) e un modello di risposta gerarchico, AIML può essere scritto in diversi formati, tra cui le espressioni S di Lisp , JSON, YAML o un formato di testo strutturato. Le specifiche AIML 2.0 includono anche una rappresentazione alternativa: un ibrido di file flat e XML chiamato AIML Intermediate Format (descritto nella sezione seguente).   
La modifica di AIML inevitabilmente compromette la sua semplicità originale. L'aggiunta di più tag e funzionalità rende la lingua più difficile da comprendere. AIML 2.0 è un tentativo di affrontare le carenze riscontrate dai maestri di bot negli ultimi due decenni, continuando a perseguire l'obiettivo originale della semplicità. Queste specifiche AIML 2.0 sono ampiamente compatibili con le versioni precedenti di AIML 1.0 e degli standard precedenti. Le nuove funzionalità si basano sul linguaggio originale in modo tale che i concetti possano essere organizzati pedagogicamente in livelli principiante, intermedio e avanzato.

### Panoramica del sistema AIML

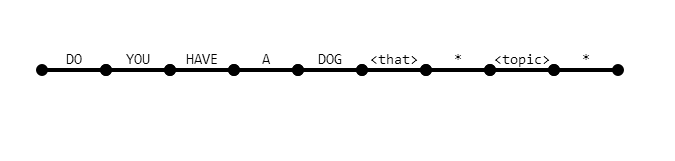
AIML definisce una relazione tra tre entità: un chiacchierone umano chiamato client, un autore di bot chat umano chiamato botmaster e il robot o bot stesso. In generale un botmaster può creare più robot e ogni bot può avere più client. Un sistema come Pandorabots prevede più botmaster, più bot e più client. Un sistema AIML incorporato in un dispositivo consumer potrebbe avere solo un bot e un client. Lo standard AIML non specifica il numero di bot, botmaster o client (tranne per il fatto che definire AIML significa che dobbiamo parlarne almeno uno di ciascuno). I dettagli sulla gestione di più bot, botmaster e client sono lasciati all'implementazione.   
Bisogna comunque prestare attenzione a gestire lo stato di ciascun bot e di ogni sessione client.

Lo standard AIML non specifica dove o come sono definite proprietà, insiemi, mappe, sostituzioni e predicati. Questo è un dettaglio di implementazione lasciato al progettista dell'interprete. I valori possono essere immessi tramite un'interfaccia utente, salvati in file di testo o in un database o in qualsiasi altro formato tra cui XML e JSON, purché l'interprete possa leggerli all'avvio del bot.

### Corrispondenza del modello AIML

Ogni categoria AIML è specificata in modo univoco da un modello di input, tale modello e modello di argomento. Ricordare che se e / o non vengono specificati, si presume che abbiano un valore predefinito \*. Un percorso di modello è definito come una sequenza collegata di nodi, in cui i nodi sono collegati da bordi etichettati con le parole. La sequenza di parole in un percorso di modello è specificata come le parole dal modello di input, seguite dal simbolo, dalle parole in quel modello, seguite dalle parole nel modello di argomento.   
La Figura 1 mostra il percorso del motivo per una categoria

<category>   
<pattern>DO YOU HAVE A DOG</pattern>   
<template>Can your dog be my pet too?</template>   
</category>



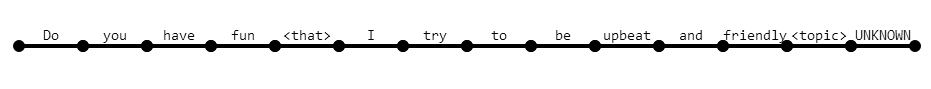
**Figura 1.***Percorso del modello per una categoria.*

L’interprete AIML crea un oggetto chiamato Graphmaster leggendo i file AIML, costruendo un percorso modello per ogni categoria e inserendo il percorso in un grafico diretto e radicato. Alla fine di ogni percorso, Graphmaster contiene un collegamento al modello AIML per la categoria associata.

Dato un input specifico per il bot, l'interprete AIML crea un percorso di input, simile a un percorso di pattern, contenente l'input normalizzato, l'ultima risposta del bot (il valore di), anch'esso normalizzato, e l'argomento normalizzato. La Figura 3 illustra un esempio di un percorso di input risultante dal frammento di conversazione:

**Robot** : cerco di essere ottimista e amichevole.

**Umano** : ti diverti?

Ai fini di questo esempio, l'argomento è "sconosciuto".   


**Figura 3.***Percorso di input con <that> e <topic>*

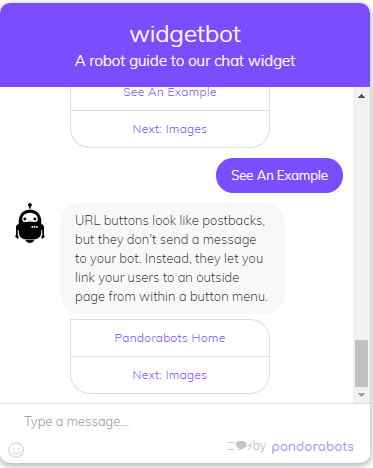
L’algoritmo di corrispondenza del modello AIML cerca nel Graphmaster una corrispondenza del percorso di input. La ricerca procede in una sequenza in profondità. Quando la ricerca in un ramo del grafico non riesce a trovare una corrispondenza, l'algoritmo di ricerca torna all'ultimo nodo con rami inesplorati e cerca quelli.

Il carattere jolly **#** può corrispondere a **zero o più** parole dal percorso di input.   
Il carattere jolly **\_** può corrispondere a **una o più** parole dal percorso di input.   
Per una corrispondenza esatta delle parole, la parola successiva nel percorso di input deve essere identica (fino all'invarianza del caso) con la parola che identifica il ramo.   
Una corrispondenza di set consuma anche una o più parole, come un carattere jolly, ma la sequenza di parole deve essere un membro del set AIML denominato (vedere Set e mappe in AIML 2.0).   
Il carattere jolly **^** può corrispondere a **zero o più** parole dal percorso di input.   
Infine, il carattere jolly **\*** può corrispondere a **una o più** parole.   
Ai fini della corrispondenza, i simboli speciali e i percorsi di input e pattern sono trattati come parole esatte.   
Se non viene trovata alcuna corrispondenza nel grafico, l'interprete dovrebbe restituire una stringa predefinita come "Non ho una risposta per questo", specificata dal botmaster.

#### Tag Pulsante URL

Il tag URL <button> consente alla risposta <template> di visualizzare un pulsante su cui il client (utente) può fare clic nell'app di messaggistica per avviare un sito Web. Il sito Web può essere visualizzato nell'app di messaggistica o in un'app per browser esterna.   
Il pulsante URL utilizza due tag secondari: <testo> e <url>. Il sottotag <text> contiene il testo da visualizzare nel collegamento e il tag <url> specifica il collegamento da aprire.   
Esempio:

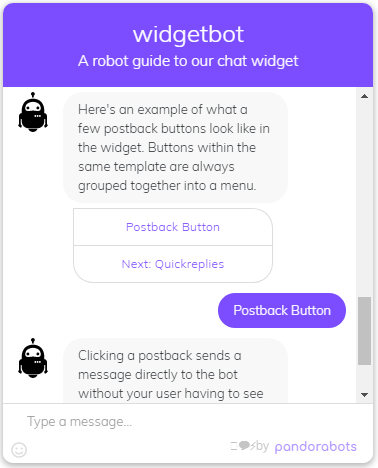
<button>  
  <text>Pandorabots Home</text>  
  <url>https://www.pandorabots.com </url>  
</button>



#### Pulsante postback

Il "postback" nel pulsante Postback descrive come il testo del pulsante viene rinviato al bot, come se l'utente avesse appena digitato o pronunciato il testo del pulsante. In effetti, la funzione di questo pulsante non impone alcun vincolo alla risposta del cliente: possono fare clic sul pulsante, digitare o dire la risposta o inserire qualcosa di completamente diverso. Resta responsabilità dell'autore del bot creare la logica in AIML per gestire queste risposte in modo diverso, se lo si desidera.   
Esempio:   
la coppia di categorie

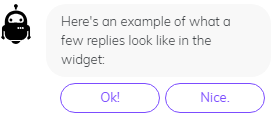
<category>   
  <pattern>EXAMPLE POSTBACK</pattern>   
  <template>   
    Here's an example of what a few postback buttons look like in the widget.   
    <button>   
      <text>Postback Button</text>   
      <postback>POSTBACK BUTTON</postback>   
    </button>   
  </template>   
</category>

display:   


#### risposte

Il tag <reply> implementa una finestra di dialogo che consente all'utente di inviare una delle diverse possibili risposte al bot, scegliendo rapidamente toccando la selezione o, facoltativamente, digitandolo o salvandolo. Un tag <reply> è molto simile a un pulsante postback, tranne per il fatto che offre più di una scelta. Il tag <reply> contiene un elenco di espressioni di testo / postback. Ognuno visualizza il testo associato e, se l'utente sceglie tale opzione, rimanda il postback associato al bot.   
Esempio:   
la coppia di categorie

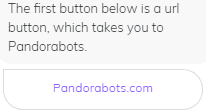
<category>   
  <pattern>REPLY EXAMPLE</pattern>   
  <template>   
    Here's an example of what a few reply buttons look like in the widget. Replies within the same template are always grouped together beneath your last message.   
    <reply>   
      <text>Ok!</text>   
      <postback>POSTBACK</postback>   
    </button>   
  </template>   
</category>   
<category>   
  <pattern>POSTBACK</pattern><that>REPLIES \*</that>   
  <template>   
    Quick replies work just like postbacks, and send a message directly to your bot without the user having to see it. That's how you hit this category!   
    <reply>   
      <text>Ok!</text>   
      <postback>Nice!</postback>   
    </button>   
  </template>   
</category>



#### collegamento ipertestuale

Agli albori di AIML, quando la maggior parte dei bot veniva visualizzata su pagine Web, consentivamo a <modello> AIML di contenere un mix di tag HTML e AIML. Per consentire al bot di visualizzare un collegamento ipertestuale dell'immagine o di stilizzare la risposta, il botmaster potrebbe includere tag HTML come <a>, <img> o <b> come parte della risposta del bot.   
Sfortunatamente le piattaforme di messaggistica istantanea mancano di un linguaggio comune standard come HTML per implementare una tale decorazione di testo banale. Per questo motivo AIML 2.1 specifica i tag Rich Media per <immagine>, <video> e <hyperlink>.   
Esempio:

<template>   <link>  
    <text>Pandorabots.com</text>  
    <url>https://www.pandorabots.com </url>  
  </link> </template>



#### Immagine

<image>www.png|jpg|gif</image>

#### video

<video>www.mp4</video>

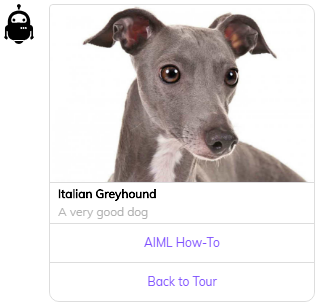
#### Carta

Un tag <card> avvolge altri elementi: un tag immagine , un certo numero di pulsanti , nonché un titolo e un sottotitolo , entrambi contenenti testo. Il risultato è un menu che contiene tutti questi elementi rich media.

<card>   
  <image>www.png</image>   
  <title>Card</title>   
  <subtitle>Subtitle</subtitle>   
  <button>   
    <text>Option 1</text>   
    <postback>DO OPTION 1</postback>   
  </button>   
  <button>   
    <text>Option 2</text>   
    <postback>DO OPTION 2</postback>   
  </button>   
  <button>   
    <text>Option 3</text>   
    <postback>DO OPTION 3</postback>   
  </button>   
</card>

Esempio:

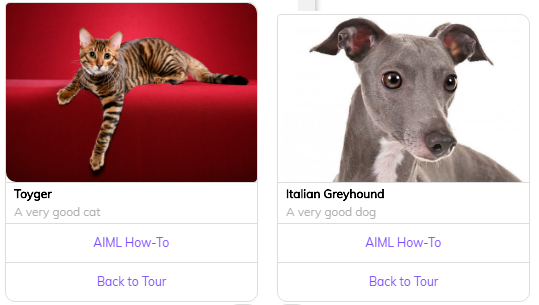
<card>   
  <image>www.png</image>   
  <title>Italian Greyhound</title>   
  <subtitle>A very good dog</subtitle>   
  <button>   
    <text>AIML How-To</text>   
    <postback>HOW TO</postback>   
  </button>   
  <button>   
    <text>Back To Tour</text>   
    <postback>RESUME TOUR</postback>   
  </button>   
</card>



#### Giostra

Un tag <carousel> avvolge un certo numero di carte per creare un menu di scorrimento organizzato in diverse sezioni.

<carousel>   
  <card>   
    ...   
  </card>   
</carousel>



#### Ritardo

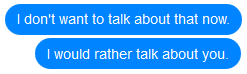
I ritardi sono utili per introdurre una pausa tra le parti di una risposta per una lettura più semplice o per simulare il tempo impiegato da un essere umano a leggere e digitare una risposta. Sono facili da usare: basta avvolgere un tag <delay> per il numero di secondi che si desidera attendere.

<delay>3</delay>

#### Diviso

Le divisioni fanno esattamente ciò che dicono: dividono un messaggio in più parti. Questi verranno visualizzati all'utente come messaggi separati, che possono essere combinati con ritardi per distanziare le risposte di grandi dimensioni. Per usarli, basta inserire un <split /> tra il contenuto che si desidera separare.

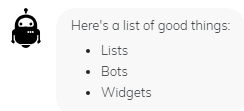
<category>   
  <pattern>TEST SPLIT</pattern>   
  <template>   
    I don't want to talk about that now.   
    <split/>   
    I would rather talk about you.   
  </template>   
</category>



#### Elenco puntato

Visualizza un elenco di punti elenco puntato.

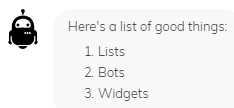
<template>   
  Here's a list of good things:   
  <list>   
    <item>Lists</item>   
    <item>Bots</item>   
    <item>Widgets</item>   
  </list>   
</template>



#### Lista ordinata

Visualizza un elenco numerato di elementi.

<template>   
  Here's a list of good things:   
  <olist>   
    <item>Lists</item>   
    <item>Bots</item>   
    <item>Widgets</item>   
  </olist>   
</template>



Link:

1. <https://chatbotslife.com/rule-based-standalone-aiml-chatbots-chatbots-part-2-f5dca9f15956>
2. <https://www.tutorialspoint.com/aiml/aiml_introduction.htm?source=post_page--------------------------->
3. <https://uxplanet.org/humanizing-chatbots-by-designing-conversational-uis-a92b25fa3e1a>
4. <https://medium.com/pandorabots-blog/aiml-tutorial-creating-a-context-aware-multi-functional-chatbot-e5e82c027a6a>
5. <https://studylibit.com/doc/838035/costruzione-di-basi-di-conoscenza-aiml-per-chatter-bot-a-...>
6. <https://home.pandorabots.com/home.html>