

INTELIGENCIA ARTIFICIAL

E.T.S. de Ingenierías Informática y de
Telecomunicación

Práctica 1 ANEXO 2

Extensión del lenguaje AIML

ANEXO 2

Durante la presentación de la práctica se ha hecho una descripción de las componentes más importantes del lenguaje AIML 2.0. En este anexo, se describen algunas funcionalidades adicionales que se han incluido en este lenguaje para aumentar su expresividad.

Extensión del lenguaje AIML

El lenguaje AIML 2.0 tiene algunas restricciones a la hora de poder trabajar con conjuntos y diccionarios, así también para poder utilizar patrones `<pattern>` con expresiones regulares. Es por ello que se ha desarrollado en un Trabajo Fin de Grado (realizado por Benjamín Vega Herrera) una versión extendida que contempla nuevas características que hacen más llevadera la programación en este lenguaje.

2.1 Conjuntos en AIML

Los conjuntos en AIML 2.0 se acceden mediante la etiqueta `<set>`, pero solamente se pueden utilizar dentro del `<pattern>` y no en el `<template>`, ya que se confundiría con la etiqueta `<set>` para guardar valores en variables. Para darle mayor flexibilidad se ha desarrollado una versión extendida del lenguaje que permite:

- Cargar en una variable los valores de un conjunto:

```
<set var="valores"><readset>telefono</readset></set>
```

- Guardar en un conjunto un nuevo valor:

```
<addtoset>  
  <name>datos</name>  
  <key>email</key>  
</addtoset>
```

donde `<name>` hace referencia al nombre del fichero que almacena el “set”, en este caso, el fichero se llama “datos.txt” y está en la carpeta “set”, y `<key>` hace referencia al nuevo valor que se desea incluir, en este caso, “email”. Es un conjunto, por tanto, si “email” ya existiera, no se incluye este valor de nuevo en el “set”.

- Eliminar en un conjunto un valor:

```
<removefromset>  
  
<name>datos</name>  
  
<key>email</key>  
  
</removefromset>
```

donde `<name>` hace referencia al nombre del fichero que almacena el “set”, en este caso, el fichero se llama “datos.txt” y está en la carpeta “set”, y `<key>` hace referencia al nuevo valor que se desea incluir, en este caso, “email”.

2.2 Diccionarios en AIML

Los diccionarios en AIML 2.0 se acceden mediante la etiqueta `<map>`. Esta etiqueta puede utilizarse solamente dentro del `<template>`. Por ejemplo, puedo intentar guardar en una variable local llamada “`eltel`” el valor asociado al diccionario *telefono* utilizando *Sanchez* como clave de la siguiente manera (las dos formas son equivalentes):

```
<set var=“eltel”><map name=“telefono”>Sanchez</map></set>  
  
<set var=“eltel”><map><name>telefono</name>Sanchez</map></set>1
```

Lamentablemente los diccionarios en AIML 2.0 son fijos, es decir, no se pueden modificar desde dentro del AIML. Es por ello que se ha desarrollado una versión extendida del lenguaje que permite:

- Cargar en una variable las claves de un diccionario (así luego puedo averiguar si existe una cierta clave en un “map”):

```
<set var=“claves”><readkeys>telefono</readkeys></set>
```

- Guardar en un diccionario un nuevo par (clave, valor):

```
<addtomap>  
  
<name>telefono</name>  
  
<key>Perez</key>  
  
<value>111222333</value>
```

¹ Esta segunda versión resulta útil cuando el nombre del diccionario se encuentra almacenado en una variable.

`</addtomap>`

donde `<name>` hace referencia al nombre del fichero que almacena el “map”, en este caso, el fichero se llama “telefono.txt” y está en la carpeta “map”, `<key>` hace referencia a la nueva clave que se desea incluir, en este caso, “Perez”, y `<value>` es el valor que se asocia con esa clave. Si la clave ya existe, esta operación no modifica el “map”.

- Modificar el valor asociado a una clave de un diccionario:

```
<modifymap>
  <name>telefono</name>
  <key>Sanchez</key>
  <value>111222333</value>
</modifymap>
```

- Borrar del diccionario una par (clave, valor):

```
<removefrommap>
  <name>telefono</name>
  <key>Sanchez</key>
</removefrommap>
```

en este último caso, no es necesario indicar el campo `<value>`.

2.3 Nuevas opciones para `<pattern>`

Se ha agregado la posibilidad de utilizar algunas expresiones regulares más comunes dentro de la etiqueta `<pattern>`. A continuación se listan y ejemplifican cada una de ellas:

- **Palabras opcionales:** Muchas veces nos interesa poder permitir la aparición de una palabra opcional en una expresión regular (en otros lenguajes se utiliza el símbolo “?” seguido de lo que se quiere hacer opcional). En AIML se ha optado por incluir la palabra opcional entre paréntesis. Por ejemplo, el patrón

`<pattern>tengo (muchas) posibilidades</pattern>`

se enlazará correctamente con las frases: “tengo posibilidades” y con “tengo muchas posibilidades”. Incluso se pueden utilizar varias veces en una misma frase:

`<pattern>un numero al azar (uno) (dos) (tres)</pattern>`

se enlazará correctamente con “un numero al azar”, “un numero al azar uno”, “un numero al azar uno dos”, “un numero al azar uno dos **tres**”, “un numero al azar uno tres” y “un numero al azar dos tres”.

- **Subconjunto de palabras:** En algunos casos el uso del “*” es demasiado amplio, y nos gustaría poder restringirlo a un conjunto más pequeño. En AIML se ha optado por utilizar corchetes para indicar esto, de forma similar a otros lenguajes. Por ejemplo, si solo quiero permitir que puedan aparecer dos palabras (si o no) escribiría el siguiente patrón:

`<pattern>el me dijo que [si no] queria</pattern>`

esto permite que se enlace correctamente solo dos posibles frases: “el me dijo que si quería” o bien “el me dijo que no quería”.

- **Prefijos y sufijos:** Muchas veces nos resulta útil poder colocar la raíz de un verbo e indicar que todas las posibles conjugaciones también se redirijan a la misma porción de código. Para ello se ha incluido la posibilidad de utilizar sufijos mediante el carácter “+”.

`<pattern>^ guard+ ^</pattern>`

`<template>Guardando dato</template>`

De esta manera se puede llegar al trozo de código asociado a ese patrón cuando el usuario ingrese diferentes frases, ej: “por favor guardame este dato”, “no te olvides de guardar este dato”, “ya me estas guardando este dato”, etc. De forma similar se pueden utilizar prefijos anteponiendo el “+” a la palabra deseada.



UNIVERSIDAD
DE GRANADA

Departamento de Ciencias de la
Computación e Inteligencia Artificial

/ UGR / decsai

<pattern>^ +ndo ^</pattern>

<template>Arreando que es gerundio</template>