

Formularze i gramatyki dla przemiennej inicjatywy

Pozwalają na przypisanie wartości więcej niż jednemu polu podczas pojedynczego wejścia użytkownika.

Formularz o przemiennej inicjatywie posiada:

- element `<initial>`, który jest zdefiniowany dla całego formularza
- gramatykę, która również jest zdefiniowana dla całego formularza – tag `<grammar>` musi być wewnątrz formularza, lecz nie w obiekcie wejściowym (polu `<field>`).

Przykład

Prosty formularz umożliwiający wybór serwisu.

```
<form>
  <field name="selection">
    <prompt>Please choose News, Weather, or Sports.</prompt>
    <grammar type="application/x-nuance-gsl">
      [news weather sports]
    </grammar>
  </field>
</block>
  <submit next="select.jsp"/>
</block>
</form>
```

Przykład cd.

Załóżmy, że chcemy umożliwić wybór jednego, dwóch lub trzech serwisów (w dowolnej kolejności).

Zamiast pola selection użyjemy trzech oddzielnych pól, będących slotami. Każdy z nich będzie polem logicznym, któremu przypisze się wartość true, gdy użytkownik poprosi o ten serwis.

Gramatyka posiada tzw. zasady gramatyczne (grammar rules):

- 1) pierwsza zasada na najwyższym poziomie: główna zasada (tutaj o nazwie **Request**)
 - pierwsza litera nazwy – wielka,
 - definiuje co będzie rozpoznawane, czyli reprezentuje całe wejście użytkownika,
 - w ciele tej zasady zamiast trzech opcji do wyboru, trzykrotnie występuje słowo (tutaj **Service**), będące nazwą kolejnej zasady, ta nazwa kolejnej zasady również rozpoczyna się wielką literą;
- 2) druga zasada (**Service**) rozpoznaje wybór, zawiera trzy nazwy serwisów z przypisanymi im komendami definiującymi sloty (nazwa serwisu może zacząć się od „the”).

```
<form>
  <grammar type="application/x-nuance-gsl">
    <![CDATA[
      Request
      (
        ?[ (i'd like) (tell me) ]
        Service
        ?and ?Service
        ?and ?Service
        ?please
      )
      Service
      (
        ?the
        [
          news    { <news true> }
          weather { <weather true> }
          sports  { <sports true> }
        ]
      )
    ]]>
  </grammar>
```

```
<initial>
  <prompt>
    What service would you like?
  </prompt>
</initial>
```

```
<field name="news" type="boolean"> </field>
<field name="weather" type="boolean"> </field>
<field name="sports" type="boolean"> </field>
```

```
<filled mode="any">
  <if cond="news"><prompt>You chose the news service.</prompt> </if>
  <if cond="weather"><prompt>You chose the weather service.</prompt> </if>
  <if cond="sports"><prompt>You chose the sports service.</prompt> </if>
  <clear/>
</filled>
</form>
```

UWAGI:

- Formularz ma inną strukturę – zaczyna się od gramatyki, która może rozpoznawać różnorodne wejścia użytkownika, np.: "News, weather.", "Tell me sports and news.", "I'd like news and sports and weather."
- Połączenie gramatyki z polami formularza jest określone następująco:

```
news      { <news true> }  
weather   { <weather true> }  
sports    { <sports true> }
```

Fragmety zawarte w nawiasach {} są komendami (definicjami slotów), które przechowują wartości true w slotach. Tutaj sloty nazywają się: news, weather, sports.

- Potem jest element **<initial>**, który wykonywany jest zawsze jako pierwszy w formularzu. Zawiera on podpowiedź otwierającą i może również wykonywać inne działania, jak inicjowanie zmiennych.
- Następnie są trzy pola typu boolean nazwane news, weather, sports. Ponieważ te nazwy odpowiadają nazwom slotów zdefiniowanych w gramatyce, interpreter automatycznie przypisze zawartość slotów tym polom. W ten sposób komendy gramatyki przechowują wartości w tych polach.
- W końcu jest element **<filled>**, który jest wykonywany po tym, jak interpreter zakończy przetwarzać wejście użytkownika. **Ten element posiada atrybut mode ustawiony na any.** Domyślnie element **<filled>** wykonywany jest tylko wtedy, gdy wszystkie pola formularza są wypełnione. W tym przypadku oznaczałoby to, sytuację wyboru wszystkich trzech serwisów. Aby umożliwić wykonywanie elementu, gdy tylko dowolne pole formularza jest wypełnione, należy ustawić atrybut mode na any.
- W przykładzie element **<filled>** nie dostarcza danych dla serwera - tylko odtwarza podpowiedź, aby potwierdzić co zostało rozpoznane, a następnie używa **<clear/>** aby wyzerować wszystkie pola formularza, tak aby można było go ponownie użyć dla innego wejścia.