

6. Programmieren in VoiceXML

Programmieren in VoiceXML

Donnerstag 01. 07., 08. 07. und 15. 07. 2004

6. Programmieren in VoiceXML

Richtziel:

Erlernen möglicher Verwendung von Markup-Programmiersprachen für Modellierung der natürlichsprachlichen Mensch-Computer Interaktion.

Schwerpunkte der Unterricht:

Grundlagen der Computerlinguistik

Modellierung und Simulation sprachlicher Prozesse auf dem Computer

Grundlagen der Mensch-Computer Interaktion

Natürlichsprachliche Kommunikation, Discourse und Dialog

Funktionale Eigenschaften und Blockstruktur des Dialogsystems

Einführung in die Problematik der Sprachein- und -ausgabe

Prinzipien der Dialogsteuerung, symbolische Darstellungen der Dialogführung

Rolle der Dialogmodelle und Dialogmodellierung

6. Programmieren in VoiceXML

Eigenschaften von Markup Programmiersprachen

HTML, AIML, XML und VoiceXML Sprachen

Aufbau (Struktur) und Elemente der Sprache VoiceXML

Merkmale von VoiceXML Programmen

Beispiele

Erstellung von Programmen in VoiceXML unterschiedlicher Komplexität

6. Programmieren in VoiceXML

Grundstruktur des VoiceXML Programms:

```
<?xml version="1.0">
<vxml version="2.0">
<var name="xyz" expr="16"/>      <!-- Variablendef. -->
<form id="form_name">
  <block>                          <!-- "feste" Ausgabe -->
    <prompt> Text oder <value expr="hi"/> </prompt>
    <goto next="#next_module"/>
  </block>
  <field name="answer">           <!-- Spracheingabefeld -->
    <grammar> yes{yes} | no{no} </grammar>
    <catch event="noinput">      <!-- Ereignisse -->
      <prompt> Hey, don't sleep! </prompt>
    </catch>
```

6. Programmieren in VoiceXML

```
<catch event="nomatch">
  <prompt> say 'yes' or 'no' </prompt>
</catch>
<filled>      <!-- ausgeführt nach Bedingung -->
  <if cond="answer=='yes'">
    <prompt> Answer 1 </prompt>
  <else/>
    <prompt> Answer 2 </prompt>
  </if>
</filled>
</field>
</form>
</vxml>
```

6. Programmieren in VoiceXML

Einfache Dialogführung (simple initiative):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vxml version="2.0">
<!-- Weather example -->

<catch event="noinput">
  <prompt> Hey, don't sleep! </prompt>
</catch>
<catch event="exit">
  <prompt> Exit! </prompt>
  <exit/>
</catch>
<link event="exit">
  <grammar> exit | quit | q </grammar>
</link>
```

6. Programmieren in VoiceXML

```
<form id="weather_info">
  <block>Welcome to the weather information service.</block>
  <field name="state">
    <prompt>What state?</prompt>
    <grammar src="state.gram" type="application/x-jsgf"/>
    <catch event="help">
      <grammar> [please] help [me] [please] </grammar>
      Please speak the state for which you want the weather.
    </catch>
  </field>
  <field name="city">
    <prompt>What city?</prompt>
    <grammar src="city.gram" type="application/x-jsgf"/>
    <catch event="help">
      <grammar> [please] help [me] [please] </grammar>
```

6. Programmieren in VoiceXML

Please speak the city for which you want the weather.

```
</catch>
```

```
</field>
```

```
<block>
```

```
<prompt> Transmission of program control to the weather  
information system </prompt>
```

```
<prompt> to provide the information about the weather in <value  
expr="city$.utterance"/> ,
```

```
<value expr="state$.utterance"/> </prompt>
```

```
<submit next="/servlet/weather" namelist="city state"/>
```

```
</block>
```

```
</form>
```

```
</vxml>
```


6. Programmieren in VoiceXML

Programmausführung:

Welcome to the weather information service.

What state?

Georgia

What city?

Tbilisi

Unrecognized input

What city?

Macon

Transmission of program control to the weather information system to provide the information about the weather in Macon, Georgia

6. Programmieren in VoiceXML

Gemischte Dialogführung (mixed initiative):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vxml version="2.0">
<!-- Weather example 2 -->

<catch event="noinput">
  <prompt> Hey, don't sleep! </prompt>
</catch>
<catch event="exit">
  <prompt> Exit! </prompt> <exit/>
</catch>
<link event="exit">
  <grammar> exit | quit | q </grammar>
</link>
```

6. Programmieren in VoiceXML

```
<form id="weather_info2">
```

```
<grammar src="citystat.gram" type="application/x-jsgf"/>
```

```
<!-- Caller can't barge in on today's advertisement. -->
```

```
<block>
```

```
  <prompt bargein="false">
```

```
    Welcome to the weather information service.
```

```
    <audio src="http://www.online-ads.example.com/wis.wav"/>
```

```
  </prompt>
```

```
</block>
```

6. Programmieren in VoiceXML

```
<initial name="start">
  <prompt>
    For what city and state would you like the weather?
  </prompt>
  <help>
    Please say the name of the city and
    state for which you would like a weather report.
  </help>
  <!-- If user is silent, reprompt once, then
    try directed prompts. -->
  <noinput count="1"> <reprompt/></noinput>
  <noinput count="2"> <reprompt/>
  <assign name="start" expr="true"/></noinput>
</initial>
```

6. Programmieren in VoiceXML

```
<field name="state">
  <prompt>What state?</prompt>
  <help>
    Please speak the state for which you want the weather.
  </help>
</field>
```

```
<field name="city">
  <prompt>Please say the city in <value expr="state"/>
    for which you want the weather.</prompt>
  <help>
    Please speak the city for which you want the weather.
  </help>
```

6. Programmieren in VoiceXML

```
<filled>
  <!-- Most of our customers are in LA. -->
  <if cond="city == 'Los Angeles' && state == undefined">
    <assign name="state" expr="'California'"/>
  </if>
</filled>
</field>

<field name="go_ahead" type="boolean" modal="true">
  <prompt>Do you want to hear the weather for <value
    expr="city"/>, <value expr="state"/>? </prompt>
  <filled>
    <if cond="go_ahead">
      <prompt bargein="false">
```

6. Programmieren in VoiceXML

```
<audio src="http://www.online-ads.example.com/wis2.wav"/>
</prompt>
<submit next="/servlet/weather" namelist="city state"/>
<prompt> Transmission of program control to the weather
        information system to provide the information
        about the weather in <value expr="city$.utterance"/>,
        <value expr="state$.utterance"/> </prompt>
</if>
<clear namelist="start city state go_ahead"/>
</filled>
</field>
</form>
</vxml>
```

6. Programmieren in VoiceXML

Programmausführung:

Welcome to the weather information service. Buy Joe's Spicy Shrimp Sauce.
For what city and state would you like the weather?

Uh, California.

Please say the city in California for which you want the weather.

San Francisco, please.

Do you want to hear the weather for San Francisco, California?

No

For what city and state would you like the weather?

Los Angeles.

Do you want to hear the weather for Los Angeles, California?

Yes

Don't forget, buy Joe's Spicy Shrimp Sauce tonight!

Mostly sunny today with highs in the 80s. Lows tonight from the low 60s...

6. Programmieren in VoiceXML

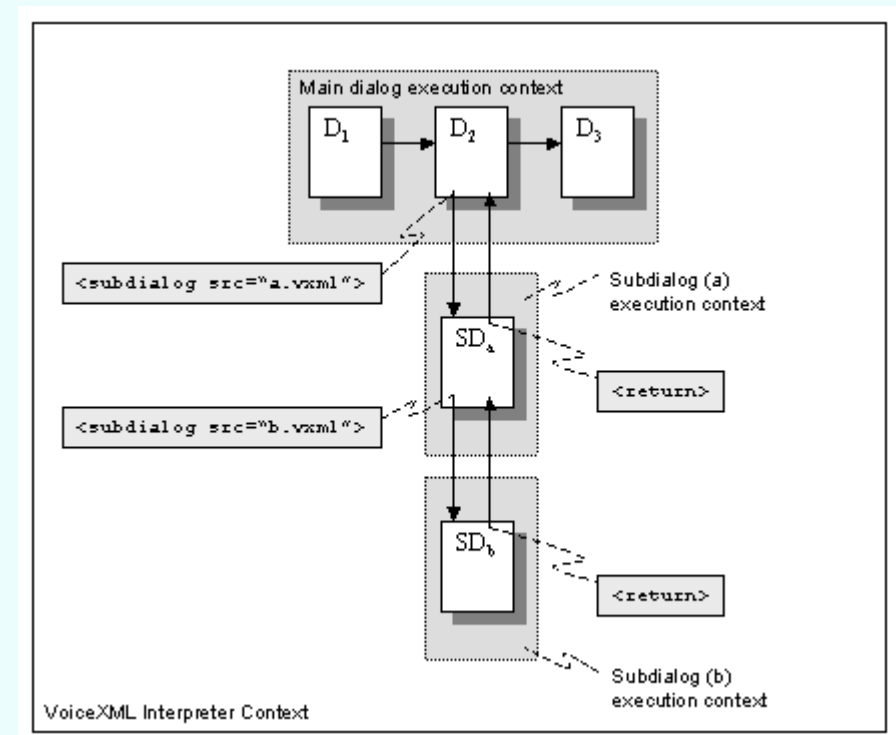
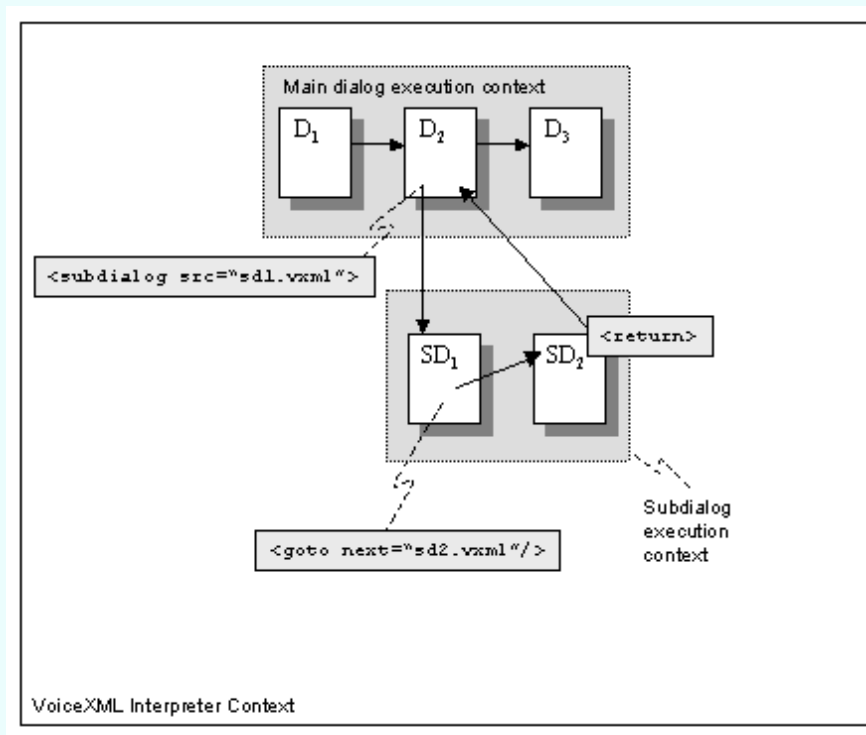
Subdialogs und ihre Verwendung

Prinzip:

A subdialog is a mechanism for decomposing complex sequences of dialogs to better structure them, or to create reusable components. For example, the solicitation of account information may involve gathering several pieces of information, such as account number, and home telephone number. A customer care service might be structured with several independent applications that could share this basic building block, thus it would be reasonable to construct it as a subdialog. This is illustrated in the example below. The first document, app.vxml, seeks to adjust a customer's account, and in doing so must get the account information and then the adjustment level. The account information is obtained by using a subdialog element that invokes another VoiceXML document to solicit the user input. While the second document is being executed, the calling dialog is suspended, awaiting the return of information. The second document provides the results of its user interactions using a <return> element, and the resulting values are accessed through the variable defined by the name attribute on the <subdialog> element.

6. Programmieren in VoiceXML

Serielle und geschachtelte Reihung (Ordnung)



6. Programmieren in VoiceXML

Beispiel:

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="billing_adjustment">
    <var name="account_number"/>
    <var name="home_phone"/>
    <subdialog name="accountinfo" src="acct_info.vxml#basic">
      <filled>
        <!-- Note the variable defined by "accountinfo" is
              returned as an ECMAScript object and it contains two
              properties defined by the variables specified in the
              "return" element of the subdialog. -->
        <assign name="account_number" expr="accountinfo.acctnum"/>
        <assign name="home_phone" expr="accountinfo.acctphone"/>
      </filled>
    </subdialog>
```

6. Programmieren in VoiceXML

```
<field name="adjustment_amount" type="currency">
  <prompt>
    What is the value of your account adjustment?
  </prompt>
  <filled>
    <submit next="/cgi-bin/updateaccount"/>
  </filled>
</field>
</form>
</vxml>
```

6. Programmieren in VoiceXML

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="basic">
    <grammar src="number.gram" type="application/x-jsgf"/>
    <field name="acctnum">
      <prompt> What is your account number? </prompt>
      <filled>
        <prompt> Your account-nr is <value expr="acctnum$.utterance"/> </prompt>
      </filled>
    </field>
    <field name="acctphone">
      <prompt> What is your home telephone number? </prompt>
      <filled>
        <!-- The values obtained by the two fields are supplied
              to the calling dialog by the "return" element. -->
        <return namelist="acctnum$.utterance acctphone$.utterance"/>
      </filled>
    </field>
  </form>
</vxml>
```

6. Programmieren in VoiceXML

Beispiel mit mehrfachen Bestätigungen

```
<?xml version="1.0"?>
<vxml version="2.0">

  <var name="state"/>
  <var name="city"/>
  <var name="conf_result"/>
  <catch event="noinput">
    <prompt> Hey, don't sleep! </prompt>
  </catch>

  <form id="subdialog_demo">
    <block>
      <prompt> State to City Assignment </prompt>
    </block>
```

6. Programmieren in VoiceXML

```
<field name="city_input">
  <prompt> Say the name of the requested city </prompt>
  <grammar src="citystat.gram" type="application/x-jsgf"/>
  <filled>
    <assign name="city" expr="city_input$.utterance"/>
    <prompt> Did you say <value expr="city"/> ? </prompt>
    <subdialog name="confirm" src="Ssubd.vxml#basic">
      <filled>
        <assign name="conf_result" expr="confirm.conf"/>
      </filled>
    </subdialog>
    <if cond="conf_result=='no'">
      <clear name="conf_result"/>
    </if>
  </filled>
</field>
```

6. Programmieren in VoiceXML

```
<block name="state_input_request">
  <prompt> Do you want to give the state ? </prompt>
  <subdialog name="confirm2" src="Ssubd.vxml#basic2">
    <filled>
      <assign name="conf_result" expr="confirm2.conf2"/>
    </filled>
  </subdialog>
  <if cond="conf_result=='no'">
    <if cond="city == 'Los Angeles' || city == 'San Francisco'">
      <assign name="state" expr="'California'"/>
    <elseif cond="city == 'Boston'"/>
      <assign name="state" expr="'Massachusets'"/>
    <elseif cond="city == 'Dallas' || city == 'Houston'"/>
      <assign name="state" expr="'Texas'"/>
    </if>
    <goto next="#closing"/>
  </if>
```


6. Programmieren in VoiceXML

```
</block>
```

```
<field name="state_input">
```

```
  <prompt> Say the name of the state </prompt>
```

```
  <grammar src="citystat.gram" type="application/x-jsgf"/>
```

```
  <filled>
```

```
    <assign name="state" expr="state_input$.utterance"/>
```

```
    <prompt> Did you say <value expr="state"/> ? </prompt>
```

```
    <subdialog name="confirm" src="Ssubd.vxml#basic">
```

```
      <filled>
```

```
        <assign name="conf_result" expr="confirm.conf"/>
```

```
      </filled>
```

```
    </subdialog>
```

```
    <if cond="conf_result=='yes'">
```

```
      <goto next="#closing"/>
```

```
    <else/>
```

```
      <clear namelist="conf_result"/>
```

6. Programmieren in VoiceXML

```
        </if>
        </filled>
    </field>
</form>

<form id="closing">
    <block>
        <prompt> Your assignment is : <value expr="city"/> -
            <value expr="state"/>
        </prompt>
        <prompt> Good bye ! </prompt>
    </block>
</form>
</vxml>
```

6. Programmieren in VoiceXML

```
<?xml version="1.0"?>
<vxml version="2.0">

<!-- Modul von Subdialogs - enthält 2 Arten von Subdialogs -->

  <form id="basic">
    <field name="conf">
      <prompt> Confirm please the data answering "yes" or "no" </prompt>
      <grammar> yes{yes} | y{yes} | no{no} | n{no} </grammar>

      <filled>
        <return namelist="conf"/>
      </filled>

    </field>
  </form>
```

6. Programmieren in VoiceXML

```
<form id="basic2">
  <field name="conf2">
    <prompt> Answer please "yes" or "no" </prompt>
    <grammar> yes{yes} | y{yes} | no{no} | n{no} </grammar>

    <filled>
      <return namelist="conf2"/>
    </filled>

  </field>
</form>
</vxml>
```

6. Programmieren in VoiceXML

Aufruf Nr. 1:

State to City Assignment

Say the name of the requested city

Dallas

Did you say Dallas ?

Confirm please the data answering "yes" or "no"

yes

Do you want to give the state ?

Answer please "yes" or "no"

yes

Say the name of the state

Texas

Did you say Texas ?

Confirm please the data answering "yes" or "no"

yes

Your assignment is : Dallas - Texas

Good bye !

6. Programmieren in VoiceXML

Aufruf Nr. 2:

State to City Assignment

Say the name of the requested city

Houston

Did you say Houston ?

Confirm please the data answering "yes" or "no"

yes

Do you want to give the state ?

Answer please "yes" or "no"

no

Your assignment is : Dallas - Texas

Good bye !

6. Programmieren in VoiceXML

Aufruf Nr. 3:

State to City Assignment

Say the name of the requested city

Los Angeles

Did you say Los Angeles ?

Confirm please the data answering "yes" or "no"

no

State to City Assignment

Say the name of the requested city

Los Angeles

Did you say Los Angeles ?

Confirm please the data answering "yes" or "no"

yes

Do you want to give the state ?

Answer please "yes" or "no"

yes

6. Programmieren in VoiceXML

Say the name of the state

California

Did you say California ?

Confirm please the data answering "yes" or "no"

y

Your assignment is : Los Angeles - California

Good bye !

6. Programmieren in VoiceXML

Aufruf Nr. 4:

State to City Assignment

Say the name of the requested city

Los angeles

Unrecognized input

Say the name of the requested city

Los Angeles

Did you say Los Angeles ?

Confirm please the data answering "yes" or "no"

no

State to City Assignment

Say the name of the requested city

Los Angeles

Did you say Los Angeles ?

Confirm please the data answering "yes" or "no"

yes

6. Programmieren in VoiceXML

Do you want to give the state ?

Answer please "yes" or "no"

no

Your assignment is : Los Angeles - California

Good bye !

6. Programmieren in VoiceXML

Aufruf Nr. 5:

State to City Assignment

Say the name of the requested city

Hey, don't sleep!

Boston

Did you say Boston ?

Confirm please the data answering "yes" or "no"

yes

Do you want to give the state ?

Answer please "yes" or "no"

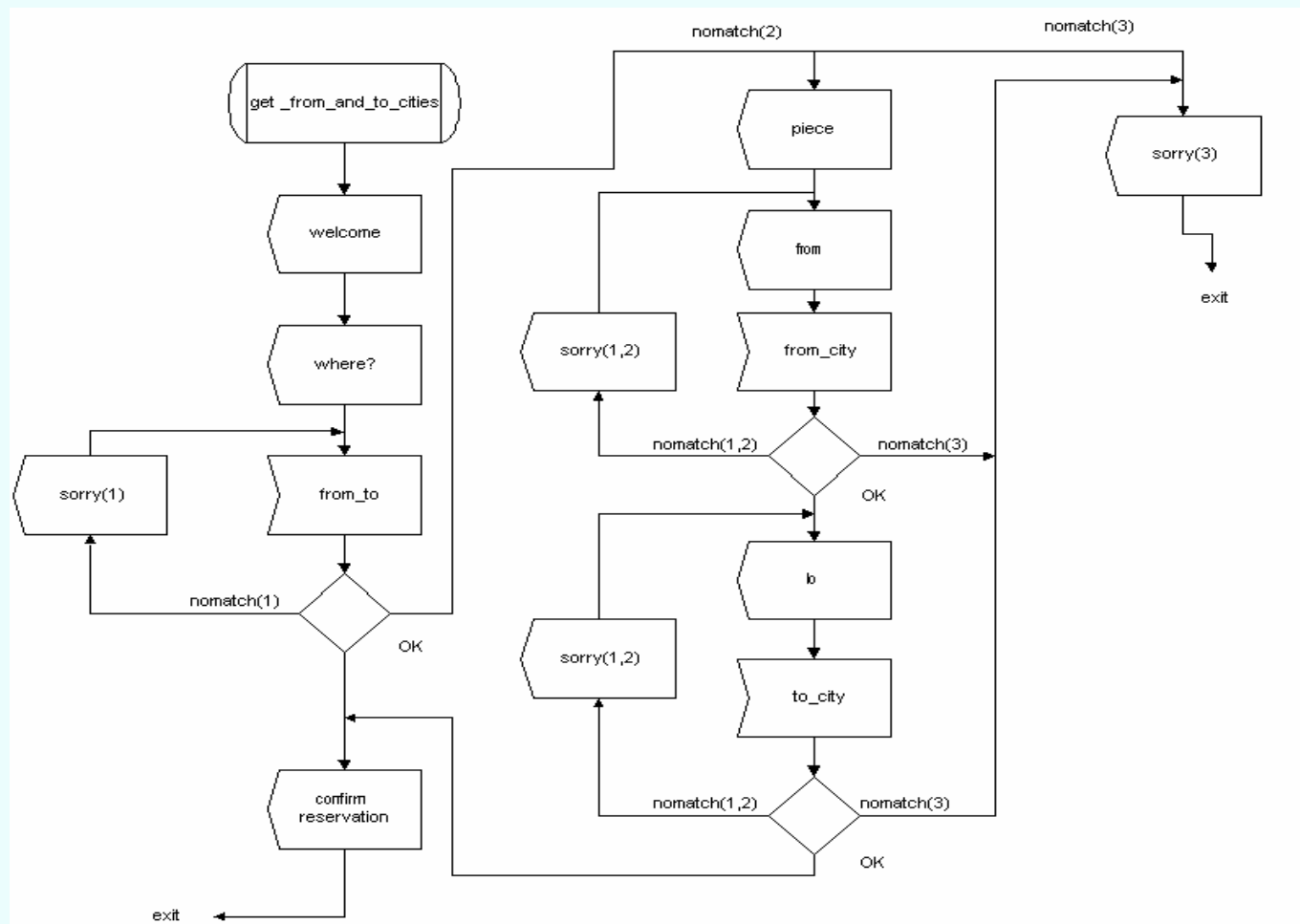
no

Your assignment is : Boston - Massachusets

Good bye !

6. Programmieren in VoiceXML

Wie nach dem Dialogmodel programmieren?



6. Programmieren in VoiceXML

Dialog mit einem Reservationssystem:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<vxml version="1.0">

<!-- mixed initiative dialog, filled and counters example -->

<link event="again">
  <grammar>[please] [all] again [please] | [please] once more
  [please]</grammar>
</link>

<nomatch>
  Sorry, I did not understand you.
  <reprompt/>
</nomatch>

<form id="get_from_and_to_cities">
  <grammar src="cities.jsg"/>
```

6. Programmieren in VoiceXML

```
<catch event="again">
  <prompt> OK, once more!</prompt>
  <clear namelist="bypass_init from_city to_city"/>
  <reprompt/>
</catch>

<nomatch count="2">
  I am sorry, I still do not understand you. Valid cities are Paris,
    Prague and Oslo
</nomatch>
<nomatch count="3">
  I did not understand again. I am giving up. Bye. <exit/>
</nomatch>

<block> Welcome to the airplane ticket reservation system. </block>

<initial name="bypass_init">
  Where do you want to fly from and to?
  <nomatch count="1">
    Sorry, I did not understand you. Say something like "from Oslo to
      Praha".
  </nomatch>
</initial>
```

6. Programmieren in VoiceXML

```
<nomatch count="2">
  I am sorry, I still don't understand.
  I will ask you for information one piece at a time.
  <assign name="bypass_init" expr="true"/>
  <reprompt/>
</nomatch>
</initial>

<field name="from_city" slot="from">
  <grammar src="city.jsg"/>
  From which city are you leaving?
  <prompt count="2"> Tell me from which city you are leaving, please
  </prompt>
</field>

<field name="to_city" slot="to">
  <grammar src="city.jsg"/>
  To which city do you want to fly?
  <prompt count="2"> Tell me to which city you want to fly, please
  </prompt>
</field>
```

6. Programmieren in VoiceXML

```
<block>
  <prompt>
    A ticket from <value expr="from_city"/> to <value expr="to_city"/>
    is reserved for you.
  </prompt>
</block>
```

```
<filled namelist="from_city">
  <prompt>
    Your departure city is <value expr="from_city"/>.
  </prompt>
</filled>
```

```
<filled namelist="to_city">
  <prompt>
    Your arrival city is <value expr="to_city"/>.
  </prompt>
</filled>
```


6. Programmieren in VoiceXML

```
<filled>
  <if cond="from_city==to_city">
    <prompt>
      Sorry, you can't fly from <value expr="from_city"/> to
      <value expr="to_city"/>!
    </prompt>
    <prompt> Please repeat your order. </prompt>
    <clear namelist="bypass_init from_city to_city"/>
  </if>
</filled>
</form>
</vxml>
```

6. Programmieren in VoiceXML

Ein Beispiel der Dialogführung

System: Guten Tag. Hier ist die automatische Intercityauskunft. Welche Information benötigen Sie?

Benutzer: *Ich möchte morgen früh nach Berlin.*

System: Sie wollen am Freitag den 26. Juni zwischen 6 und 12 Uhr nach Berlin fahren? Wo wollen Sie abfahren?

Benutzer: *Von Regensburg.*

System: Sie wollen von Regensburg abfahren?

Benutzer: *Ja.*

System: Nach Berlin Hauptbahnhof fahren Sie ab Regensburg um 6 Uhr 54. Ankunft in Berlin Hauptbahnhof um 13 Uhr 16. Ich wiederhole: Abfahrtszeit um 6 Uhr 54. Ankunft um 13 Uhr 16. Soll ich die Verbindung wiederholen?

Benutzer: *Nein, danke. Auf Wiederhören.*

System: Auf Wiederhören.

6. Programmieren in VoiceXML

Programmrealisation:

```
<?xml version="1.0"?>
<vxml version="2.0">

  <var name="source_city"/>
  <var name="goal_city"/>
  <var name="source_time"/>
  <var name="goal_time"/>
  <var name="via_city"/>
  <var name="best_erg"/>
  <catch event="noinput">
    <prompt> Schlafe nicht, stehe auf ! </prompt>
  </catch>

  <form id="IC_demo">
    <block>
      <prompt> Intercity Auskunft </prompt>
      <prompt> Guten Tag. Hier ist die automatische Intercityauskunft.
      </prompt>
      <prompt> Welche Information benötigen Sie ? </prompt>
      <goto next="#eingabe_1"/>
    </block>
  </form>
</vxml>
```

6. Programmieren in VoiceXML

```
</block>
</form>

<form id="eingabe_1">
  <field name="ein_1">
    <grammar src="IC_Auskunft.gram" type="application/x-jsgf"/>
    <filled>
      <assign name="goal_city" expr="ein_1"/>
      <prompt> Sie wollen am Freitag den 4. Juli zwischen 6 und 12 Uhr
        nach <value expr="goal_city"/> fahren ?
      <subdialog name="bestaet" src="IC2.vxml#basic">
        <filled>
          <assign name="best_erg" expr="bestaet.best"/>
        </filled>
      </subdialog>
      <if cond="best_erg=='nein'">
        <clear name="best_res"/>
        <prompt> Ich verstehe Ihnen nicht. Wiederholen Sie, bitte, die
          Angaben. </prompt>
        <prompt> Welche Information benötigen Sie ? </prompt>
        <goto next="#eingabe_1"/>
      </if>
    </filled>
  </field>
</form>
```

6. Programmieren in VoiceXML

```
<!-- Call "Auswahl_goal_city" -->
<prompt> goal_city = <value expr="goal_city"/> </prompt>
<goto next="#eingabe_2"/>
</filled>
</field>
</form>

<form id="eingabe_2">
  <field name="ein_2">
    <grammar src="IC_Auskunft.gram" type="application/x-jsgf"/>
    <prompt> Wo wollen Sie abfahren ? </prompt>
    <filled>
      <assign name="source_city" expr="ein_2"/>
      <prompt> Sie wollen von <value expr="source_city"/> abfahren ?
      </prompt>
      <subdialog name="bestaet" src="IC2.vxml#basic">
        <filled>
          <assign name="best_erg" expr="bestaet.best"/>
        </filled>
      </subdialog>
      <if cond="best_erg=='nein'">
        <clear name="best_res"/> <goto next="#eingabe_2"/>
      </if>
    </filled>
  </field>
</form>
```

6. Programmieren in VoiceXML

```
</if>
<!-- Call "Auswahl_source_city" -->
<prompt> source_city = <value expr="source_city"/> </prompt>
<goto next="#antwort_1"/>
</filled>
</field>
</form>
```

```
<form id="antwort_1">
  <block>
    <prompt> Nach <value expr="goal_city"/> Hbf fahren Sie ab
      <value expr="source_city"/> um ... </prompt>
    <goto next="#eingabe_3"/>
  </block>
</form>
```

```
<form id="eingabe_3">
  <block name="ein_3">
    <prompt> Soll ich die Verbindung wiederholen ? </prompt>
    <subdialog name="bestaet" src="IC2.vxml#basic">
      <filled>
        <assign name="best_erg" expr="bestaet.best"/>
      </filled>
    </subdialog>
  </block>
</form>
```

6. Programmieren in VoiceXML

```
</filled>
</subdialog>
<if cond="best_erg=='ja'">
  <goto next="#antwort_1"/>
</if>
<prompt> Benötigen Sie eine weitere Information ? </prompt>
<subdialog name="bestaet" src="IC2.vxml#basic">
  <filled>
    <assign name="best_erg" expr="bestaet.best"/>
  </filled>
</subdialog>
<if cond="best_erg=='ja'">
  <goto next="#eingabe_4"/>
</if>
<goto next="#abschied"/>
</block>
</form>

<form id="eingabe_4">
  <block>
    <prompt> Welche weitere Information benötigen Sie ? </prompt>
    <goto next="#eingabe_1"/>
  </block>
</form>
```

6. Programmieren in VoiceXML

```
</block>
</form>

<form id="abschied">
  <block>
    <prompt> Auf Wiederhören ! </prompt>
  </block>
</form>
</vxml>
```

Subdialog:

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="basic">
    <field name="best">
      <grammar> ja{ja} | j{ja} | nein{nein} | n{nein} </grammar>
      <filled>
        <return namelist="best"/>
      </filled>
    </field>
  </form>
</vxml>
```


6. Programmieren in VoiceXML

Grammatik für die Spracheingabe:

```
grammar IC_Auskunft;                                // name

public <main> = <frage1> | <frage2> | <abschied>;
<frage1> = <bitte> [<tag>] [<zeit>] <prep> <ort>;
<frage2> = <prep> <ort>;
<bitte>   = ich moechte;
<tag>    = heute | morgen;
<zeit>   = frueh | vormittags | nachmittags | abends;
<ort>    = Berlin | Muenchen | Regensburg;
<prep>   = von | nach;
<abschied> = [danke] auf wiederhoeren;
```

6. Programmieren in VoiceXML

Gesteuerter Dialog (directed form):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vxml version="2.0">

<form id="get_card_ident">
  <block> Jetzt geben Sie die Art, Nummer und das Gültigkeitsdatum Ihrer
    Kreditkarte ein.</block>
  <field name="card_type">
    <prompt count="1">Welche Kreditkarte haben Sie?
  </prompt>
  <prompt count="2"> Geben Sie bitte den Typ Ihrer Kreditkarte
    ein.</prompt>
  <grammar> <!-- Da wird eine inline-Grammatik definiert -->

    <rule id="rule1" scope="public">
      <one-of>
        <item>visa</item>
```

6. Programmieren in VoiceXML

```
<item>master
    <count number="optional">card</count></item>
<item>amex</item>
<item>american express</item>
</one-of>
</rule>
</grammar>
<help>  Sagen Sie Visa, Mastercard, oder American Express.
</help>
</field>

<field name="card_num" type="digits">
    <!-- Grammatik für type="digits" ist eingebaut. -->
<prompt count="1"> Wie ist die Nummer Ihrer Karte?</prompt>
<prompt count="2"> Geben Sie Ihre Kartenummer ein!</prompt>
<catch event="help">
```

6. Programmieren in VoiceXML

```
<if cond="card_type=='amex' || card_type=='american express'"> Geben Sie die  
fünfzehnstellige Nummer Ihrer Karte ein.
```

```
<else/>
```

```
    Geben Sie die sechzehnstellige Nummer Ihrer Karte ein.
```

```
</if>
```

```
</catch>
```

```
<filled>          <!-- Eingabe der Kartennummer -->
```

```
<if cond="(card_type=='amex' || card_type=='american express')  
    & & card_num.length != 15">
```

```
    Die Nummer der American Express Karte muss  
    15 Ziffern haben!
```

```
    <clear namelist="card_num"/>
```

```
    <throw event="nomatch"/>
```

```
<elseif cond="card_type != 'amex' & &  
    card_type != 'american express' & &  
    card_num.length != 16"/>
```

6. Programmieren in VoiceXML

Die Nummer der Mastercard und Visa Karte muss
16 Ziffern haben!

```
<clear namelist="card_num"/>  
<throw event="nomatch"/>
```

```
</if>
```

```
</filled>
```

```
</field>
```

```
<field name="expiry_date" type="digits">
```

```
  <prompt count="1"> Bis wann ist Ihre Karte gültig?</prompt>
```

```
  <prompt count="2"> Geben Sie das Gültigkeitsdatum Ihrer Kreditkarte  
    ein!</prompt>
```

```
  <help> Sagen Sie, bis wann ist Ihre Kreditkarte gültig ist, zum Beispiel  
    null sieben null vier. </help>
```

```
</filled> <!-- Das Gültigkeitsdatum überprüfen -->
```

```
  <var name="mm"/>
```

```
  <var name="i" expr="expiry_date.length"/>
```

6. Programmieren in VoiceXML

```
<if cond="i == 3">
  <assign name="mm" expr="expiry_date.substring(0,1)"/>
<elseif cond="i == 4"/>
  <assign name="mm" expr="expiry_date.substring(0,2)"/>
</if>
<if cond="mm == '' || mm < 1 || mm > 12">
  <clear namelist="expiry_date"/>
  <throw event="nomatch"/>
</if>
</filled>
</field>

<field name="confirm" type="boolean">
  <prompt>
    Sie haben die Nummer <value expr="card_num"/>
```

6. Programmieren in VoiceXML

der `<value expr="card_type"/>` Kreditkarte, die bis `<value expr="expiry_date"/>` gültig ist, eingegeben. Sind diese Angaben korrekt?

```
</prompt>
```

```
<filled>
```

```
  <if cond="confirm">
```

```
    <submit next="place_order.asp"
```

```
      namelist="card_type card_num expiry_date"/>
```

```
  </if>
```

```
  <clear
```

```
    namelist="card_type card_num
```

```
    expiry_date acknowledge"/>
```

```
</filled>
```

```
</field>
```

```
</form>
```

```
</vxml>
```

6. Programmieren in VoiceXML

Die Ausführung (Interpretierung) dieses Formulars führt zum folgenden Dialog des Benutzers mit dem Computer:

...

System: Jetzt geben Sie die Art, Nummer und Gültigkeitsdatum Ihrer Kreditkarte ein.

System: Welche Kreditkarte haben Sie?

Anrufer: Ja, einen Moment, ich schaue mal...

System: Entschuldigung, was haben Sie gesagt?

(plattformspezifische implizite Antwort)

System: Geben Sie bitte den Typ Ihrer Kreditkarte ein. *(zweiter Prompt)*

Anrufer: Was soll ich machen? Ich brauch' Hilfe. *(Bitte um Hilfe)*

System: Welche Kreditkarte haben Sie? Sagen Sie bitte entweder Visa, Mastercard oder American Express.

Anrufer: Hmm.., ich hab' ne Mastercard. *(„Hmm“ wird von der Plattform ignoriert)*

System: Und wie ist die Nummer Ihrer Karte?

Anrufer: Acht sieben null vier eins – – Moment...

System: Ich habe nicht verstanden, was Sie gesagt haben. Können Sie die Nummer Ihrer Karte bitte wiederholen? *(kurze Pause)* Sagen Sie jetzt bitte einfach die Nummer

6. Programmieren in VoiceXML

an. Sie können die Nummer aber auch mit den Tasten Ihres Telefons eingeben.
(Eingabe mittels Spracherkennung bzw. DTMF)

Anrufer: 8 7 0 1 4 3 5 7 6 6 3 0 0 5 3 2 (Eingabe mit DTMF)

System: Bis wann ist Ihre Karte gültig ?

Anrufer: Null neun null drei

System: Sie haben eine Mastercard mit der Nummer 8 7 0 1 4 3 5 7 6 6 3 0 0 5 3 2 , die bis 0 9 0 3 gültig ist, eingegeben. Sind diese Angaben korrekt?

Anrufer: Ja.

...

6. Programmieren in VoiceXML

Ware in einem e-shop telephonisch einkaufen:

a) Das Wurzeldokument (eshop_wurzel.vxml):

```
?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Wurzeldokument für e-Shop-System - Service -->
<vxml version="2.0">

  <var name="warenart"/>    <!-- globale Variablen -->
  <var name="warentyp"/>
  <var name="hersteller"/>
  <var name="katalognummer"/>
  <var name="kundennummer"/>
  <nomatch>
    Entschuldigung, was haben Sie gesagt? <reprompt/>
  </nomatch>
  <catch event="noinput">
```

6. Programmieren in VoiceXML

```
<prompt> Können Sie bitte eine Auswahl treffen? </prompt>
</catch>
<catch event="again">
  <prompt> OK, dann noch einmal! </prompt>
  <!-- <clear namelist=""/> -->
  <reprompt/>
</catch>
<link event="again">
  <grammar>[bitte] [alles] noch einmal [bitte] |
           [bitte] noch einmal [bitte] </grammar>
</link>
<catch event="exit">
  <prompt> Vielen Dank, dass Sie die Alles-Klar AG, den Anbieter für
           kundenfreundliche Haushalts- und Elektroniklösungen, gewählt
           haben. Ich freue mich, Sie mal wieder begrüßen zu dürfen.
  </prompt>
  <!-- "leerer" Ausgang (ohne Bestellung) -->
```

6. Programmieren in VoiceXML

```
<exit/>
</catch>
<link event="exit">
  <grammar> ausgehen | verlassen | quit | schluss | beenden | abschliessen
    | schliessen | raus
  </grammar>
</link>
<link event="help">
  <grammar> [bitte] hilfe | hilf [mir] [bitte] </grammar>
</link>

<form id="kundennr"> <!-- Eingabe der Kundennummer -->
  <field name="nr">
    <grammar type="application/srgs+xml"
      src="/grammars/digits.grxml"/>
    <prompt> Geben Sie bitte Ihre Kundennummer ein.
    </prompt>
```

6. Programmieren in VoiceXML

```
<filled>
  <assign name="kundennummer" expr="nr"/>
</filled>
</field>
</form>

<form id="kontonr">
<!-- Formular kann als subdialog aufgerufen werden -->
  <var name="kontonummer"/>
  <var name="blz"/>
  <field name="knr">
    <grammar type="application/srgs+xml"
      src="/grammars/digits.grxml"/>
    <prompt> Geben Sie bitte Ihre Kontonummer ein.
  </prompt>
  <filled>
    <assign name="kontonummer" expr="knr"/>
```

6. Programmieren in VoiceXML

```
</filled>
</field>
<field name="blz_nr">
  <grammar type="application/srgs+xml"
    src="/grammars/digits.grxml"/>
  <prompt> Geben Sie bitte die Bankleitzahl ein.
</prompt>
  <filled>
    <assign name="blz" expr="blz_nr"/>
  </filled>
  <return namelist="kontonummer blz"/>
</field>
</form>
. . .

<!-- Weitere global-definierte Formulare -->
</vxml>
```

6. Programmieren in VoiceXML

b) Startseite, mit der das System aufgerufen wird (`eshop_willk.vxml`) :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vxml version="2.0 application="eshop_wurzel.vxml">

<!-- Beispiel: In einem e-Shop einkaufen -->
<!-- Eröffnungs- und Initialisierungsprogramm -->

<form id="e-shop_interaction">
  <block>
    Herzlich Willkommen im Voice Voice Portal der Alles-Klar AG, dem
    Anbieter für kundenfreund-liche Elektronik- und Haushaltslösungen. Ich bin
    ein Sprachdienst mit dem Sie telefonisch Produkte aus unserem breiten
    Sortiment bestellen können. Sprechen Sie bitte deutlich, damit ich Fehler
    bei der Spracherkennung vermeiden kann. Ich biete Ihnen jetzt einige
    Produktgruppen an. Wählen Sie bitte eine daraus aus und sagen Sie diese
    bitte deutlich an.

    <goto next="eshop_main_menu.vxml"/>
  </block>
</form>
```

6. Programmieren in VoiceXML

```
</block>  
</form>  
</vxml>
```

c) Hauptmenü (eshop_main_menu.vxml) :

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<vxml version="2.0 application="eshop_wurzel.vxml">  
<!-- Hauptmenü für einen e-Shop -->  
  
<menu id="hauptmenue">  
  <catch event="help">  
    <prompt> Wählen Sie bitte eine der folgenden Produktgruppen: <enumerate/>  
  </prompt>  
</catch>  
  
  <prompt> Wählen Sie bitte aus dem folgenden Sortiment:  
    <enumerate/>
```


6. Programmieren in VoiceXML

```
</prompt>
<choice next="/eshop_menu_ct.vxml">
  Computertechnik
</choice>
<choice next="/eshop_menu_czb.vxml">
  Computerzubehör
</choice>
<choice next="/eshop_menu_soft.vxml">
  Software
</choice>
<choice next="/eshop_menu_klima.vxml">
  Klimaanlage
</choice>
<choice next="/eshop_menu_ut.vxml">
  Unterhaltungstechnik
</choice>
<choice next="/eshop_menu_st.vxml">
```

6. Programmieren in VoiceXML

```
    Steuerungstechnik
</choice>
<choice next="/eshop_menu_hg.vxml">
    Haushaltsgeräte
</choice>
<choice next="/eshop_menu_gg.vxml">
Gartengeräte
</choice>
<choice event="exit"/>
    zurück
</choice>
<noinput>Wählen Sie bitte eine Produktgruppe aus diesem Angebot:
    <enumerate/>
</noinput>
</menu>
</vxml>
```

6. Programmieren in VoiceXML

d) Menü für die Auswahl von Haushaltgeräten (eshop_menu_hg.vxml) :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vxml version="2.0 application="eshop_wurzel.vxml">
<!-- Menü für Haushaltsgeräte -->

<menu id="menue_hg">
  <catch event="help">
    <prompt>Wählen Sie bitte eine der angebotenen Gerätegruppen:
      <enumerate/>
    </prompt>
  </catch>
  <prompt> Wählen Sie bitte eine der folgenden Gerätegruppen: <enumerate/>
  </prompt>
  <choice next="/hg_menu_he.vxml">
    Herd
  </choice>
```

6. Programmieren in VoiceXML

```
<choice next="/hg_menu_mw.vxml">
  Mikrowelle
</choice>
<choice next="/hg_menu_ks.vxml">
  Kühlschrank
</choice>
<choice next="/hg_menu_sp.vxml">
  Spülmaschine
<!-- . . . -->
<choice next="/hg_menu_sg.vxml">
  Staubsauger
</choice>
<choice next="/hg_menu_wm.vxml">
  Waschmaschine
</choice>
<choice next="/eshop_main_menu.vxml"/>
  zurück
```

6. Programmieren in VoiceXML

```
</choice>
<noinput> Wählen Sie bitte eine Gruppe aus diesem Angebot: <enumerate/>
</noinput>
</menu>
</vxml>
```

e) Menü für die Auswahl des Staubsaugers (hg_menu_sg.vxml) :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vxml version="2.0" application="eshop_wurzel.vxml">

<!-- Auswahl des Gerätes -->

<form id="auswahl_geraet">
  <field name="hersteller">
    <catch event="nomatch">
      <prompt> Der von Ihnen genannte Hersteller
```

6. Programmieren in VoiceXML

```
<value expr="hersteller$.utterance+' '/> existiert leider  
nicht. Wählen Sie bitte zwischen: <enumerate/>
```

```
</prompt>
```

```
</catch>
```

```
<catch event="help">
```

```
<prompt>Wählen Sie bitte nur einen der folgenden Hersteller:
```

```
<enumerate/>
```

```
</prompt>
```

```
</catch>
```

```
<option value="A"> AEG </option>
```

```
<option value="B1"> Bosch </option>
```

```
<option value="B2"> Braun </option>
```

```
<option value="L"> Leifheit </option>
```

```
<option value="P"> Philips </option>
```

```
<!-- ... usw. ... -->
```

```
<prompt> Wählen Sie einen der folgenden Hersteller:
```

6. Programmieren in VoiceXML

```
        <enumerate/>
</prompt>
<filled>
    <if cond="hersteller=='A'">
        <assign name="application.hersteller" expr="'AEG'"/>
<elseif cond="hersteller=='B1'">
    <assign name="application.hersteller"
        expr="'Bosch'"/>
<elseif cond="hersteller=='B2'">
    <assign name="application.hersteller"
        expr="'Braun'"/>
<!--      ...      -->
<if cond="hersteller=='P'">
    <assign name="application.hersteller"
        expr="'Philips'"/>
<!--      ...      -->
</if>
```

6. Programmieren in VoiceXML

```
</filled>
<subdialog name="typen" src="/typ_wahl.vxml"/>
  <filled>
    <submit next="/dienste/verkauf"
      namelist="application.warentyp"/>
    <!-- Annahme der Bestellung -->
  </filled>
</subdialog>
</field>
</form>
</vxml>
```

f) Auswahl des Typen (typ_wahl.vxml) :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vxml version="2.0 application="eshop_wurzel.vxml">
<!-- Menü für Typenauswahl -->
```


6. Programmieren in VoiceXML

```
<form id="typ_auswahl">
  <var result
  <block id="angebot">
    <prompt> Herzlichen Glückwunsch! Sie haben sich für die Marke
      <value expr="application.hersteller+' '/> entschieden. Eine gute
      Wahl. Wählen Sie jetzt bitte den gewünschten Typ aus dem
      folgenden Angebot: <audiosrc="/dienste/angebot/
      <value expr="application.hersteller">.wav"/>
    </prompt>
  </block>

  <field name="sg_typ">
    <catch event="nomatch">
      <prompt> Die von Ihnen genannte Marke
        <value expr="sg_typ$.utterance+' '/>
        existiert leider nicht. Wählen Sie bitte zwischen:
      <enumerate/>
    </catch>
  </field>
</form>
```

6. Programmieren in VoiceXML

```
</prompt>
</catch>
<catch event="help">
  <prompt>Wählen Sie bitte aus dem folgendem Angebot: <enumerate/>
  </prompt>
</catch>
<if cond="application.hersteller=='AEG'">
  <option value="AEG_sg1"> AG2404 </option>
  <!-- ... usw. ... -->
<elseif cond="application.hersteller=='Philips'">
  <option value="Ph_sg1"> TC318 </option>
  <option value="Ph_sg2"> TC324 </option>
  <option value="Ph_sg3"> TC424 </option>
  <option value="Ph_sg4"> TC476 </option>
  <option value="Ph_sg5"> TC524 </option>
  <option value="Ph_sg6"> TC526 </option>
  <option value="Ph_sg7"> TC628 </option>
```

6. Programmieren in VoiceXML

```
    <!--      ... usw.      ...      -->
<elseif cond="application.hersteller==' . . . '">
  <!--      ...      -->
  </if>
  <prompt> Wählen Sie bitte einen der folgenden Typen: <enumerate/>
    </prompt>
  <filled>
    <assign name="application.warentyp" expr="sg_typ">
  </filled>
</field>
</form>
</vxml>
```

6. Programmieren in VoiceXML

Beispiel der Dialogführung:

S: Herzlich Willkommen im Voice Voice Portal der Alles-Klar AG, dem Anbieter für kundenfreundliche Elektronik- und Haushaltslösungen. Ich bin ein Sprachdienst mit dem Sie telefonisch Produkte aus unserem breiten Sortiment bestellen können. Sprechen Sie bitte deutlich, damit ich Fehler bei der Spracherkennung vermeiden kann.

<<Möglichkeit, eine kurze Pause und ein Jingle einzuspielen>> Ich biete Ihnen jetzt einige Produktgruppen an. Wählen Sie bitte eine daraus aus und sagen Sie diese deutlich an: Computertechnik, Computerzubehör, Software, Klimaanlage, Unterhaltungstechnik, Steuerungstechnik, Haushaltsgeräte, Gartengeräte zurück

A: Hausjerete <simuliert eine falsche Aussprache>

S: Entschuldigung, was haben Sie gesagt haben?

A: Haushaltsgeräte <diesmal perfekte Aussprache>

S: Wählen Sie bitte eine der folgenden Gerätegruppen: Herd, Mikrowelle, Kühlschrank, Spülmaschinen, Staubsauger, Waschmaschinen, zurück

A: Einen Staubsauger.

S: Wählen Sie bitte einen der folgenden Hersteller: AEG, Bosch, Braun, Leifheit, Philips.

6. Programmieren in VoiceXML

A: Philips

S: Herzlichen Glückwunsch! Sie haben sich für die Marke Philips entschieden. Eine gute Wahl. <<Möglichkeit, Hintergrundgeräusche einer Haushaltsabteilung im Kaufhaus einzuspielen >> Jetzt wählen Sie bitte den gewünschten Typ aus dem folgenden Angebot: Firma Philips bietet Ihnen eine grosse Menge von Staubsaugern an; sie können zwischen Bodenstaubsaugern . . . <Audioausgabe> . . . Im Lager haben wir viele Modelle. Wählen Sie einen der folgenden Typen:

TC318 TC324 TC424 TC476 TC524 TC526 TC628

A: Den TC524. <<Aussprache fünfhundertvierundzwanzig>>

S: Entschuldigung, was haben Sie gesagt?

A: T C 5 2 4 <<diesmal Ziffernfolge einzeln ausgesprochen>>

S: Sie haben den Staubsauger Philips TC524 gewählt .

A: Ja.

S: Danke. Ich bereite jetzt die Bestellung vor. Bitte geben Sie Ihre Kundennummer ein.

A: eins vier vier sechs null

S: Ihre Kundennummer ist 14460. Stimmt das so ?

6. Programmieren in VoiceXML

A: Ja.

...