XISL: An Attempt to Separate Interactions from Data

Kouichi KATSURADA, Yusaku NAKAMURA, Satoshi KOBAYASHI & Tsuneo NITTA

Toyohashi University of Technology, 1-1 Hibarigaoka Tempaku-cho, Toyohashi 441-8152, Japan

katurada@tutkie.tut.ac.jp

Abstract: In this paper we outline a markup language XISL (eXtended Interaction Sheet Language) that is developed to describe multimodal interactions (MMI), and to separate the description of interactions from data in an XML document. XISL makes an XML document independent of interactions that may differ between each terminal, and so enables seamless web-browsing services to be constructed. We implemented an interpreter of XISL on lecture systems with different types of MMI and proved the viability of XISL by experiments on the systems.

Keywords: XML, multimodal interaction, XISL

1 Introduction

XML documents are now used for describing contents in many web services and are browsed through various types of terminals. Recently, some markup languages such as VoiceXML (AT&T) have been proposed to describe interactions. VoiceXML attempts to describe interactions through mobile phones, however, interactions and data are mixed in a VoiceXML document, and the modalities are limited to their target such as speech and DTMF, thus making it difficult to realize seamless services using various terminals. Since interaction facilities are different between terminals, the XML documents must be prepared for each one, but this is inconvenient because it is inefficient to prepare the same data for each terminal. We therefore propose a markup language XISL (eXtended Interaction Sheet Language) to separate the description of interactions from the data. XISL enables the data described in the XML documents to be used independently and helps us to construct seamless web-browsing services.

2 Format of XISL

Here we explain the format of XISL. The top level of an XISL document is composed of one or more elements delimited by RULE tags. Each element contains only one PATTERN item which indicates the input objects and its type by the value of an attribute MATCH. The PATTERN item has more than one EXCHANGE element, which are used to describe the response to some inputs. The EXCHANGE element is divided into two fields: the OPERATION field and the ACTION field. Multimodal input is described in the OPERATION field by means of INPUT tags. The corresponding action is described in the ACTION field by means of OUTPUT tags. In case some actions are operated synchronously, the actions are bound by the PARALLEL tag.

```
<?XML version="1.0" encoding="Shift-JIS"?>
<!DOCTYPE lecture SYSTEM "lecture.dtd">
<LECTURE>
  <START>
    Start Acceptance.
  </START>
  <ANSWER ID="1">
    <NAME> answer1 </NAME>
    <SCORE UNIT="MARKS"> 10 </SCORE>
  </ANSWER>
  <ANSWER ID="2">
    <NAME> answer2 </NAME>
    <SCORE UNIT="MARKS"> 0 </SCORE>
  </ANSWER>
  <TOTAL>
    <P> Total answers: </P>
    <ITEM> answer1: <XISL_PUT NAME="ANS1"/> </ITEM>
    <ITEM> answer2: <XISL_PUT NAME="ANS2"/> </ITEM>
  </TOTAL>
  <END>
    End of Acceptance.
  </END>
</LECTURE>
```

Figure 1: Example of XML document

```
<?XML version="1.0" encoding="Shift-JIS"?>
<!DOCTYPE lecture SYSTEM "xisl.dtd">
<XISL>
  <RULE>
    <PATTERN MATCH="START">
      <EXCHANGE>
        <OPERATION>
          <INPUT MODALITY="TOUCH" EVENT="CLICK">
        </OPERATION>
        <ACTION>
          <OUTPUT MODALITY="AGENT_SPEECH" VALUE="Input the
answers."/>
          <VAR NAME="FLAG" VALUE="1"/>
          <VAR NAME="ANS1" VALUE="0"/>
          <VAR NAME="ANS2" VALUE="0"/>
          <VAR NAME="SUM" VALUE="0"/>
        </ACTION>
      </EXCHANGE>
    </PATTERN>
  </RULE>
  <RULE>
    <PATTERN MATCH="ANSWER [@ID='1']/NAME">
      <EXCHANGE>
        <OPERATION>
          <INPUT MODALITY="TOUCH" EVENT="CLICK">
        </OPERATION>
        <ACTION>
          <IF COND="FLAG == 1">
            <EVAL EXPR="ANS1=ANS1+1"/>
          </IF>
          <PUT NAME="ANS1"/>
        </ACTION>
      </EXCHANGE>
    </PATTERN>
  </RULE>
    :
  <RULE>
    <PATTERN MATCH="END">
      <EXCHANGE>
        <OPERATION>
          <INPUT MODALITY="USER_SPEECH" EVENT="end">
        </OPERATION>
        <ACTION>
          <GET VAR="SCORE1" VALUE="ANSWER [@ID='1']/SCORE"/>
          <EVAL EXPR="SUM = ANS1 * SCORE1"/>
          <GET VAR="SCORE2" VALUE="ANSWER [@ID='2']/SCORE"/>
          <EVAL EXPR="SUM = (SUM + ANS2 * SCORE2) / (ANS1 + ANS2)"/>
          <OUTPUT MODALITY="AGENT_SPEECH"
              VALUE="Average is $SUM$ marks."/>
        </ACTION>
      </EXCHANGE>
    </PATTERN>
  </RULE>
```

Figure 2: Example of XISL document

Figure 1 and 2 show an example of XML and XISL. In this example, a teacher clicks the NAME objects displayed on a terminal with the results of a multiple choice test in hand. The system counts the number of clicked ANSWERs. Accepting the user's speech "end", the system calculates the average of the marks and outputs it by speech.

3 Implementation

We implemented an XISL interpreter and the confirmed expressiveness of XISL by describing interactions in the MMI (MultiModal Interaction) lecture system (Yamada, 2001; Ootani, 2001) which has been developed in our laboratory. We employed

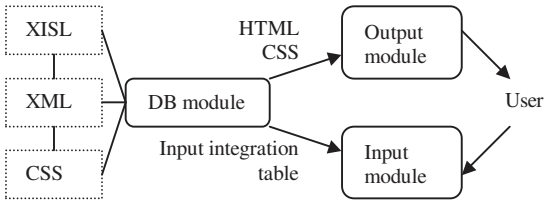

Figure 3: Data Flow in MMI Lecture System

pointing devices (mouse and touch panel), speech (TTS and speech recognition) and anthropomorphic agents as the input/output modalities.

Figure 3 shows the data flow in our system. The DB module holds the lecture content, its view, and the description of interactions as an XML document, a CSS document, and an XISL document, respectively. It produces an HTML document and an input integration table from these documents, and provides them for the input/output modules. When new data is given to the input module, the module decides its response according to the input integration table, and inform it to the DB module. We confirmed that we can properly describe the interactions in the MMI lecture system by means of XISL.

4 Concluding Remarks

We proposed a markup language XISL and confirmed its expressiveness through describing interactions in the MMI lecture system. Since we can separate the description of interactions from the data, we have only to prepare XISL documents according to the difference of terminals, thus facilitating reuse of the data described in XML documents. Our future task is to describe XISL documents for various types of applications.

References

AT&T, IBM, Lucent and Motorola, Voice XML Forum, http://www.voicexml.org/

Yamada, M. et al. (2001), Integrating modalities in multimodal interface, Proc. of the 62nd National Convention of IPSJ, 1U-04 (in Japanese).

Ootani, Y. et al. (2001), System response strategies in multimodal interface, Proc. of the 62nd National Convention of IPSJ, 1U-05 (in Japanese).