Multimodal interface of a dialogue manager

Vojtech Knezu[1], Leon Rothkrantz[2]

[1]Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University of Prague
[2]Department of Mediamatica, Faculty of Information Technology and Systems, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands (L.J.M.Rothkrantz@cs.tudelft.nl)

ABSTRACT

At Delft University of Technology there is a project running on the development of a traffic information-system. The paper presents the using of XML for a multimodal communication with an intelligent travel assistant. The task of such an assistant consists in planning a trip and supervising a traveler during his transit. It considers the Personal Intelligent Travel Assistant (PITA) and a traveler communicating with it by cellular phone, WWW and WAP communication interface. We developed an application, which transmits such communication between PITA and a user. The design and implementation of such a multimodal interface will be reported.

INTRODUCTION

The research program Seamless Multimodal Mobility (SMM) at Delft University of Technology aims at developing new public transport services for the 21st century. The services will be developed on the concept of chain mobility. In that concept, a chain manager is responsible for a perfect connection between the different transports. The traveler will be transported from door to door without spending time on finding a travel schedule or worrying about possible delays while being in transit. Communication between travelers, transport providers, and chain managers is crucial to the success of chain mobility. Such ubiquitous communication will be provided by the Personal Intelligent Travel Assistant (PITA), which is a part of the SMM program. During travel, the PITA will guide the traveler by signaling upcoming transfer points and providing information on transfer routes in the nodes. The PITA will also inform the traveler about delays, calamities, and alternative routes. Research within the PITA project focuses mainly on the question, how multimodal interaction between human and information systems can be modeled with a single dialogue manager. For an overview of multimodal communication we refer to [1].
In this paper we will report about the subproject, which was aimed at the design and simulation of an information retrieval system, which provides to its user, "a tourist arriving at Schiphol Airport", the best travel plan for public transport. As input of the system, the tourist has to provide the address or the hotel that the tourist wants to reach in the Netherlands. The tourist can use different modalities for communicating with the PITA, ranging from speech to several textual interfaces

SPEECH BASED INTERFACE

The first prototype of the PITA system we developed was provided with a speech interface and limited to a restricted domain. Current speech recognition systems allow only for a limited size of the lexicon. This would imply that only the most famous streets and hotels would be possible to recognize in the first turn. Moreover there can be ambiguity in the recognized concepts. For example "Amsterdam" is the name of the city but also the name of a famous hotel in that city. To avoid these problems the system starts with a selection dialogue. The user can choose by keywords. For the implementation of our system SpeechMania was used. SpeechMania is tool designed especially for spoken language interfaces by telephone. Later in our discussion on WAP phones we will see that the WAP interface with icons or push buttons for different services is much more appropriate.

CODING SCHEME

The dialogue manager does not use natural language or raw HTML code to represent utterances of the system and the users. Instead, it applies a higher level coding scheme for the constituting of a dialogue.

This scheme both codes the intention of an utterance as well as relevant domain information. Coded dialogues consist of turns that alternate between the system and the client. These turns consist of one or more utterances that correspond to sentences or text message. These utterances are coded as dialogue acts, which take place in a certain phase of the conversation that indicates whether the system or the client has the initiative.
Dialogues can be divided into several phases or sub-dialogues that accomplish one step in retrieving the desired information. A dialogue phase is a good indication of flow of information at a certain time in dialogue. For instance during the Information phase the traveler has the initiative and tries to describe his information need. Dialogue phases are derived by the dialogue manager and used to decide on the best strategy. The current considered phases are: Greeting, Opening, Information, Presentation, Closing and Bye. Several types of dialogue acts can be distinguished in the coding scheme. These types are used for different purposes: topic negotiation, questions, contributing information, commitments and also necessary social behavior. Information codes are used to code what type of information was transferred in every of the utterances. The coding of information is done hierarchically to facilitate automated processing of coded dialogues. The PITA coding scheme as described in [2] uses similar constructions as Prolog to code relations between information elements in dialogues. This Prolog like coding scheme has to be transferred into a coding scheme based on Extended Markup Language (XML). Integration of XML into PITA has several advantages. First, XML is becoming a widely applied standard, resulting in the portability of results of the PITA project. Secondly, general XML software can be applied within the project, like XML-parsers.
Integrating XML into PITA consists of two topics. First, the coding scheme used in PITA has to be modified to be usable with XML. The second topic consists of developing an application to facilitate communication between the PITA and a traveler based on XML.

XML CODING FOR THE PITA

The XML coding corresponds to the coding schema of PITA. It preserves the same hierarchical structure of the information codes, subcodes and attributes. Each document consists of one root <dialogue> element with an optional number of <turn> elements. A participant is identified in an attribute speaker in the turn elements. A turn consists of several utterances, with a notification of the type and phase as attributes. Codes and arguments are coded as elements. The argument elements can hold an attribute called value. The only exception is an element <between>, which can hold attributes start and end.
In the next example the dialogue manager asks the user for a departure time and place. The empty tag <about/> without any attribute expresses a request for filling a time. The multiple tags <city> in <departure_place> expresses a choice between cities.

```
Ex. <turn id=”t0001” speaker=”system”>
<utterance id=”u0003” type=”request_choice” phase=”information”>
  <time>
    <departure_time>
      <about/>
    </departure_time>
  </time>
  <location>
    <departure_place>
      <city value=”Delft”/><city value=”Amsterdam”/><city value=”Utrecht”/>
    </departure_place>
  </location>
</utterance>
</turn>
```

To summarize, if the dialogue manager requests information, it will let the tag or its attributes empty, as in the case of departure time. If the dialogue manager wants to offer a choice between more possibilities, it will enumerate all of them, as it is in departure cities. For a validation of a correct structure of a dialogue, we created a data type definition (DTD). The validation can be done with a parser.

TRANSFORMATION LAYER PROCESSING XML FOR WAP OR WWW

The transformation layer has to provide the transmission of communication between a user and the dialogue manager. The coding of communication with the manager is carried out in XML, while a common user makes use of HTML or WML (Wireless Markup Language), so it has to manage transformations between these codings. The transformation layer also has to process information data received from a user according previous communication with the dialogue manager.
The WAP and WWW have a lot in common. Both of them use markup languages to describe content. Because of the similarity we used the same approach for both modalities in the application, namely transformation style-sheets coded in XSL. As was mentioned, the demands of the application do not consist only in providing a pure transformation of an XML document according a style-sheet. It has more sophisticated tasks. For instance, in case of a request from the dialogue manager, the parameters of the user's answer have to be correspondingly inserted in XML code and delivered back to the agent. It means the request must be stored and processed after the interface layer receives the answer. We explored three different possibilities how to implement the transformation using style-sheets:

1. Web publishing framework-Cocoon

A web-publishing framework is a tool for managing requests for a published version of certain files. A published file refers to a file that may have been transformed with XSLT, and thus converted into another mark-up language like HTML, or even transformed to other formats like PDF.
One of the most progressive frameworks at the moment is Apache Cocoon. We have worked with an installation of version 1.8, based on the servlet engine Apache Jserv. Cocoon has several tag libraries, which can be used in the transformation of style-sheets. One disadvantages of using Cocoon is a serious slowness of the framework. A second disadvantage of Cocoon is the incoherence of a style-sheet code, where different tag libraries mixed.

2. Cooperation of servlets in a chain

A servlet is a Java program that dynamically extends the function of a Web server. In a chain of servlets, a request from a user is processed through several servlets. The output of one servlet is passed to an input of the other servlet, etc. The idea was to create a chain of two servlets, the first created by us, and the second servlet would be Cocoon. The task of the first servlet would be to process an XML document from the dialogue manager, and save it in memory for further processing after a user's response. The proper transformation according a style-sheet should be managed by Cocoon as the second servlet in the chain. Cocoon should transmit the result to a user. The response from the user should be delivered directly to the first servlet, which would provide the XML document and pass it to the dialogue manager. Unfortunately we could not test this procedure because the Apache Jserv servlet did not support the servlet chaining.

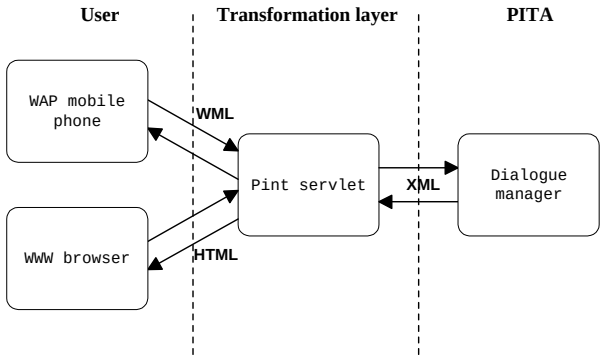

Figure 1: Data flow beteen the dialogue manager and users

3. Standalone servlet

A standalone servlet has to manage everything by itself: the preprocessing, transformation by style-sheets, processing of a response etc. For the transformation of XML to another markup language, it can reuse some Java classes, which are already done and designated for free public use. One of the other possibilities is to

use the Xalan transformation classes. This approach was chosen because of the complete control over the application and the speed. A facilitation of the development is the reuse of the transformation classes.

DESCRIPTION OF THE APPLICATION

The name of the servlet is Pint (PITA interface). As every servlet it is written in Java. During the first request of a user, the servlet creates a document tree from an initial XML document stored at the webserver. This document is transmitted by the manager. The document is a request for information, it contains several unfilled slots. The task of the servlet is to display it to the user and offer him to fill the missing information or to choose between several options. The servlet has to provide the correct filling of the received information from the user. For that reason, it saves the whole document tree between the request and the response. It creates additional ID attributes for some elements, to know where to fill the received information. The answer from the user consists of a set of variables with values. It is sent to the http address of the servlet as parameters, by either the GET or POST method. Names of the sent variables are equal to the Ids in the XML document. The values can be filled into appropriate elements in the document tree. The saved document tree is processed according to received values, they are filed to the corresponding elements. The other elements, for which no information was provided, are deleted. After processing the answer, the output XML document is saved to a file at the server. Later on, it should be transmitted to the dialogue manager and wait for a response. XML documents have to satisfy some conditions, which should be considered with creating of style-sheets. If information has to be filled by a user, the corresponding tag has to be contained in the document, but with empty values, e.g. an empty element <city/>. It expresses, that it should be displayed as input. If a user should choose between different possibilities, the document has to contain an enumeration of them. Each unfilled slot in XML should be displayed as an input. The name of the input element has to be an ID of the corresponding XML element. The value will be later filled there by the servlet. We have prepared style-sheets for WWW and WAP. They are not comprehensive, but they handle most of the tags of the coding. Their main purpose is processing of the request types of the utterances.
The running version of the servlet can be found at http://www.kbs.twi.tudelft.nl/pita/Pint. It runs on the Apache webserver with the Apache Jserv 1.1.2 servlet engine. It uses Xalan transformation classes version 1.2.2 and servlet API version 2.1.1. The servlet is developed with Java SDK version 1.3

CONCLUSION

The XML coding of information for the travel assistant is introduced. It is synoptical, easy for processing and a possible future extension of the coding should be possible without problems. Coded dialogues are well human readable, which is important for the development and testing phase of the assistant. The data type definition of the coding (DTD) is created. It can be used for checking the document's validity. In summary, the XML coding appears as appropriate for usage with the PITA.
Furthermore, the application for communication with a user through WAP and WWW interface is presented. The application is implemented as the servlet called Pint. The application uses different style-sheets for transformation between different markup languages. The advantage of the style-sheets consists in the separation of content from the transformation logic. Such style-sheets can be created also for a transformation to other markup languages, for instance VoiceXML for speech generators. We conclude that the communication between a user and the PITA can be really multimodal.

BIBLIOGRAPHY

[1] R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen and V. Zue, editors, *Survey of the state of the art in human language technology*. Cambridge University Press, 1997.

[2] R.J. van Vark, J.P.M. de Vreught, and L.J.M. Rothkrantz. "Classification of Public Transport Information Dialogues using an Information based Coding Scheme", in *Lecture Notes in Artificial Intelligence* 1236, pp.55-69, 1997.

[3] L.J.M. Rothkrantz, R.J. van Vark, A Peters, A.C. Andeweg. "Dialogue control in the ALPARON system", *Lecture Notes in Artificial Intelligence 1999.*