A Natural Language Model for Managing TV-Anytime Information in Mobile Environments

Anastasia Karanastasi, Fotis G. Kazasis, Stavros Christodoulakis

Lab. Of Distributed Multimedia Information Systems / Technical University of Crete
(MUSIC/TUC)
University Campus, Kounoupidiana, Chania, Greece
[allegra,fotis,stavros]@ced.tuc.gr

Abstract. The TV-Anytime standard describes structures of categories of digital TV program metadata, as well as User Profile metadata for TV programs. We describe a natural language model for the users to interact with the TV-Anytime metadata and preview TV programs from their mobile devices. The language utilizes completely the TV-Anytime metadata specifications and it can accommodate future metadata extensions. The interaction model does not use clarification dialogues, but it uses the user profiles to rank the possible answers in case of ambiguities, as well as TV-Anytime Metadata information and ontologies with information concerning digital TV. We describe an implementation of the language that runs on a PDA and a mobile phone and manages the metadata on a remote TV-Anytime compatible TV set.

1 Introduction

The number of digital TV channels has increased dramatically the last few years, and several industrial sectors and content producing sectors are active in defining the environment in which the TVs of the future will operate.

The TV-Anytime Forum is an association of organizations which seeks to develop specifications to enable audio-visual and other services based on mass-market high volume digital storage in consumer platforms - simply referred to as local storage [1]. These specifications target interoperable and integrated systems, from content creators/providers, through service providers, to the consumers and aim to enable applications to exploit the storage capabilities in consumer platforms. The basic architectural unit is an expanded TV set (known as a Personal Digital Recorder – PDR) capable of capturing digital satellite broadcasts according to user interests as they are described in his profile and storing them into large storage devices. The current TV-Anytime standard specifications define the structures for the metadata that can be used to describe TV programs and broadcasts, as well as for the metadata that can be used to describe the user profile. Expanded versions of the TV-Anytime architecture foresee also last mile TV-Anytime servers, Internet connection of the TV set and mobility aspects. Mobile devices (mobile phones, PDAs, etc.) in the TV-Anytime architecture can be used by a user to communicate with the home TV set not only for viewing TV programs, but also for managing the contents of the TV set (like previewing its contents, searching for content, deleting content that has been recorded for him by the TV set, etc.) and for managing his profile preferences [3].

There is a strong need for new interface paradigms that allow the interaction of naïve users with the future TV sets in order to better satisfy their dynamic preferences. The usual pc-based interfaces are not appropriate to interact with mobile devices (like mobile phones or PDAs) or with TV sets. Natural language interfaces are more appropriate interface styles for naïve users, and they can also support voice-based interactions for mobile devices.

In this paper we present a model for natural language interactions with a TV set in an environment that follows the TV Anytime specifications, both for the TV program metadata as well as for the user profile metadata. The natural language interactions are used to preview programs or summaries of programs as well as to completely manage the metadata and the programs that the TV set keeps for the user. In addition we describe an implementation of this TV-Anytime compatible natural language interaction model that works on a PDA and a mobile phone, which communicates with the TV-Anytime TV set for managing its programs and metadata and also allowing the previewing of TV programs from the mobile device.

Much research has been published in the area of natural language interfaces to interactive TV based information systems [4], [5], [6]. A well-known problem with the natural language interfaces is that user interactions may be ambiguous. Ambiguity in the natural language interfaces is a serious problem and most systems proposed in the literature lead to lengthy clarification dialogues with the user to resolve ambiguities [12]. Our own model also includes the possibility for ambiguities in the user interaction. However, our environment is more concrete since the structure imposed by the TV-Anytime specifications for the metadata limit the possibilities for ambiguities. Unlike the previous systems we do not resolve the ambiguities with clarification. Instead we can take advantage of the TV-Anytime user profile specifications in order to rank the possible interpretations and present to the user at the top position the one with the highest ranking.

The best-known dialogue systems that have been developed for digital TV and mobile environments are related to the MIINA project [8] and the Program Guide Information System of NOKIA [9]. In the context of MIINA project, a system has been developed for information retrieval from the set-top-box Mediaterminal of NOKIA. The user is allowed to insert queries for TV programs, channels, program categories and broadcast time, using a natural language. However, the natural language interaction in this model is rather simple since it is only related to the information provided by a traditional TV-Guide. The Program Guide Information System is an electronic call-in demo application offering information about television programs over the phone by allowing the user to converse with the system in natural language sentences. This system is not based on TV-Anytime metadata structures for describing the programs or the user profiles. The scope of the interaction does not include any management of the stored content or the user profiles.

The main differences between those systems and the one described in this paper is that the present system uses the TV-Anytime content and consumer metadata specifications for a complete management of TV programs and user profiles, and that the system uses additional information that exists in the TV-Anytime User Profile in order to avoid length clarification dialogues and help the user to get the most relevant answers at the top of the result list.

In section 2 of this paper the natural language model for digital TV environment is presented, along with the functionality provided and the representation of the information that the system collects from the user's input. In section 3 we present the algorithm for resolving the ambiguities instead of using clarification dialogues. In section 4 there is the analysis of the system architecture and of the modules that constitute it. Section 5 presents the implementation environment of the system and of the applications from the client side. In section 6 we present an example of a user's utterance and the actions taken by the system in order to satisfy the user's request. Finally section 7 presents the results of the system's evaluation based on user experiments and section 8 concludes by summarizing the content of this paper.

2 The Natural Language Model for the Digital TV Environment

The design of the proposed model fulfills a certain number of requirements that also determine its final functionality. Figure 1 presents the satisfied functional requirements in the form of use cases.

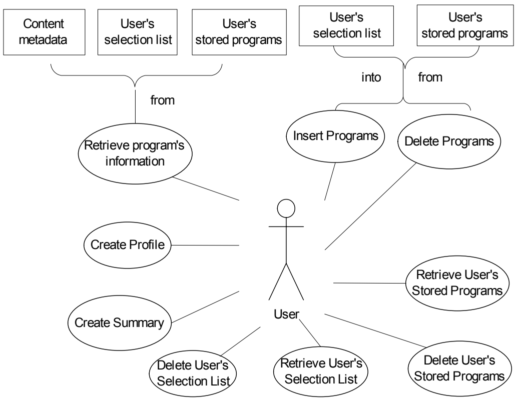

Figure 1: The use cases that the natural language system was designed to satisfy

A collection of sub-phrases was implemented and combined to constitute the user's utterance. The categories of these sub-phrases are Introduction phrases, Search phrases, Target phrases, Temporal phrases and Summary phrases.

This clustering was used to model the functionality provided by the TV-Anytime specifications. The minimal utterance consists of either a combination of an introduction phrase and a search phrase, or a combination of an introduction phrase and a target phrase. The summary phrases are individual and must be combined only with temporal phrases.

The design of the system was made taking into account two main goals. The first goal was to determine how the system should behave according to the information gathered by the user's input utterance. Based on this behavior, the second one was, to design and define the roles of each module of the system.

We created a structure for the representation of the information gathered by the user's utterance. This structure consists of three parts namely Element, Element Type and Element Value. The first structure part (Element) is used to differentiate the TV-Anytime metadata information (modeled as *TVA-properties*) from the information that directs the system to the correct management of the user's input (modeled as *flags*). The TV-Anytime information about date and time is modeled as *temporal* element. The second structure part (Element Type) is used in order to further specialize the aforementioned information and to obtain its corresponding value (the third structure part), from the user's utterance. When a user inserts an utterance into the system, it generates a feature structure [7] that follows the structure of the model.

The 'flags' element takes its value from the introduction phrases and the target phrases. The 'TVA-properties' element takes its value from the search phrases and the summary phrases and the 'temporal' element from the temporal phrases.

The feature structure can contain one, two or three elements. These three types of feature structure are:

Type 1: Markers

- E.g. I want to see what is in my selection list

This utterance consists of an **introduction phrase** (I want to see what is) and a **target phrase** (in my list). The type action of the element markers takes the value 'retrieval' (information that comes from the introduction phrase) and the element type target takes the value 'list' (information that comes from the target phrase).

Type 2: Markers – TVA-Properties

- E.g. I would like you to show me movies starring Mel Gibson

This utterance consists of an **introduction phrase** (I would like you to show me) and a **search phrase** (movies starring Mel Gibson). The type action of the element markers obtains the value 'retrieval' (information that comes from the introduction phrase), the type genre of the element TVA-properties obtains the value 'movies', the type creator takes the value 'actor' and the type name takes the value 'Mel Gibson' (information that comes from the search phrase).

Type 3: Markers – TVA-Properties – Temporal

- E.g. Insert English spoken mystery movies broadcasted at midnight into my selection list.

This utterance consists of an **introduction phrase** (Insert), a search phrase (English spoken mystery movies broadcasted), a **temporal phrase** (at midnight) and a **target phrase** (into my selection list). In this case, the type action of the element markers takes the value 'insert' (information that comes from the introduction phrase), the type target takes the value 'list', from the target phrase, the type genre of the element TVA-properties takes the values 'mystery' and 'movies', the type language takes the value 'English' and in the feature structure there is also the type dissemination value, but without value. This information comes from the search phrase. Also, in the element temporal, the type time takes the value '24' and the type time indicator takes the value 'am'. This information also comes from the search phrase.

The TV-Anytime metadata model integrates specifications for content metadata used to describe digital TV Programs in terms of various features and specifications for user preferences used to filter program metadata. The "Filtering and Search Preferences" are used to store the preferences of the users in terms of content features so that a digital TV system can filter content and provide personalization services. The FilteringAndSearchPreferences Descriptor Scheme (DS) specifies a user's filtering and/or searching preferences for audio-visual content. These preferences can be specified in terms of creation-, classification- and source-related properties of the content. The FilteringAndSearchPreferences DS is a container of CreationPreferences (i.e. Title, Creator), ClassificationPreferences (i.e. Country, Language) and SourcePreferences (i.e. DisseminationSource, DisseminationLocation). The BrowsingPreferences DS is used to specify a user's preferences for navigating and accessing multimedia content and is a container of SummaryPreferences (i.e. SummaryType, Symmary-Theme, SummaryDuration) and PreferenceCondition (i.e. Time, Place).

For the first three system functions (referring to the system functionality presented in fig. 1 above), namely the retrieval of the personalized content metadata, the management of the personalized content and the creation of the user's profile, the utterance contains in its body one or more search phrases. The system will create a TV-Anytime XML document, compatible with the UserIdentifier and the FilteringAndSearchPreferences Descriptor Schemes of the TV-Anytime metadata specification. For the forth system function, the definition of the user's preferences for the characteristics of an audio-visual content summary, the system constructs a TV-Anytime XML document, compatible with the UserIdentifier and the BrowsingPreferences Descriptor Schemes, with values in the fields of the SummaryPreferences and the PreferenceCondition (for handling the time of the summary delivery).

A user's selection list is a list of information about program's metadata that the system recommends to the user based on his preferences expressed either at his TV-Anytime profile or directly by him. Every program in this list has a status. The four possible values of this status are: undefined, toBeRecorded, recorded, toBeDeleted.

If the user wants to manage the contents of his selection list or the list of his stored contents the actions that take place are represented in figure 2:

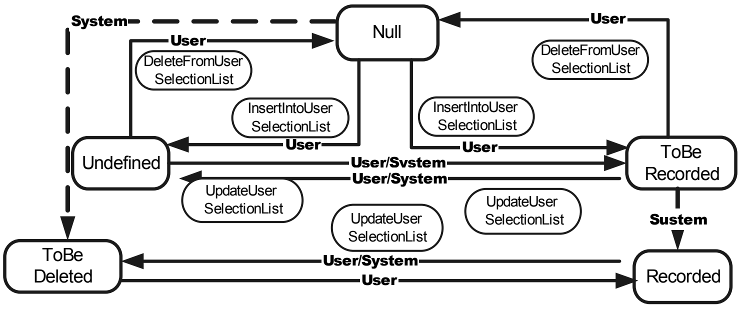

Figure 2: The state machine for managing the status of a program in the user's selection list

The user may know neither the contents nor their status of his selection list, so the system undertakes to execute the proper function on behalf of the user. The system supports the functions that are described in figure 3:

Command	Target	Action
Insert	List	InsertIntoUserSelectionList with status Undefined
Insert	PDR	If the programs were already in the selection list, then UpdateUserSelectionList with status ToBeRecorded If they were not, then InsertIntoUserSelectionList with status ToBeRecorded
Insert	-	InsertIntoUserSelectionList with status Undefined
Delete	List	DeleteFromUserSelectionList
Delete	PDR	UpdateUserSelectionList with status ToBeDeleted
Delete	-	If they were into the selection list with status Recorded then UpdateUserSelectionList with status ToBeDeleted If they were into the selection list with any status but Recorded then DeleteFromUserSelectionList
Store|Record	-	InsertIntoUserSelectionList with status ToBeRecorded

Figure 3: Functions supported by the system for managing user's selection list

3 Resolving Ambiguities – The algorithm

If the user does not specify in his utterance the TV-Anytime category he is referring to, the system tries to resolve any ambiguities by following specific steps. First, it collects word by word the sub-phrase with the ambiguities and creates a table of these words (refer to Vector of keywords in figure 4 below). Then -by using a stop list containing words with no semantic values, such as prepositions, pronouns, conjunctions, particles, articles, determiners and so on- it eliminates the words that match any word in this list. However, the system retains the existence of an 'and' or an 'or' for the optimum cross-correlation of the results. Then the system gathers the remaining words and by filtering them through its database or other existing ontologies it returns a TV-Anytime XML document that is compatible with the FilteringAndSearchPreferences DS. This descriptor scheme is the one that is used for the semantic resolving of the words. Finally, the system checks for matches in any existing TV-Anytime user's profile. This flow is represented in figure 4. The classification of the results is important in order to prioritize the user's preferences.

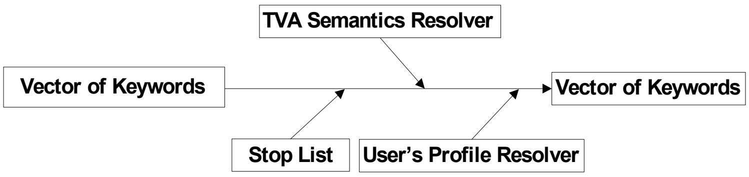

Figure 4: The system tries to give TV-Anytime semantics to the words with the ambiguities

Based on the form of the sub-phrase with the ambiguities the algorithm acts as follows:

Algorithm for ambiguities resolution

```
Subroutine: semantic resolver
    for every word with ambiguity check the stop list
    if there is no match.
         check for TVA semantics
         if there are TVA semantics add a specific weight value
             if there is a user profile
                 check from semantics from profile
                 if there is a match
                      add a weight value based on the preference value from profile to the TVA  semantics
    if there is a match
        cut the word
    return

check the words with ambiguities for an 'and' or an 'or'
if there is no match
    call semantic resolver
    check for same strings that contain words with the same semantic
    cut the rest
    search for results
   rank them based on the weight value
else
    call semantic resolver
    search for results
    group the results based on the same TVA semantic
      rank the results based on the weight value
```

An example for the case where there is no match between the stop list and the words with the ambiguities would be the sub-phrase **'… with Nicole Kidman'**. Suppose that the TV-Anytime Semantics Resolver gives the semantics shown in table 1.

Creator	**Role**
Nicole Rosselle	Actor
Mary Kidman	Director
Nicole Kidman	Producer
Nicole Kidman	Actor

Table 1: Semantics from the TV-Anytime Semantics Resolver for the words 'Nicole', 'Kidman'

Suppose now that the User's Profile Resolver gives the semantics with the corresponding reference value as the one declared in the TV-Anytime User's Profile, shown in table 2.

Creator	**Role**	**Pval**
Nicole Rosselle	actor	80
Nicole Kidman	producer	50
Nicole Kidman	actor	70

Table 2: Semantics based on the TV-Anytime User's Profile for the words 'Nicole', 'Kidman'

The results in this case, based on the above, are shown in table 3.

	Creator	**Role**
1	Nicole Kidman	Actor
2	Nicole Kidman	Producer

3	Nicole Rosselle	Actor
4	Mary Kidman	Director

Table 3: Results after resolving the ambiguity of the sub-phrase '… with Nicole Kidman'

An example for the case where there is a match between the stop list and the words with the ambiguities and there is an 'and' or an 'or' between the words with the ambiguities would be the sub-phrase **'… with Frank and English'**. Suppose that the TV-Anytime Semantics Resolver gives the semantics shown in table 4.

Creator	**Role**	**Language**
Frank English	actor	english
Holger Franke	director	
Frank… 66 results	various	

Table 4: Semantics from the TV-Anytime Semantics Resolver for the words 'Frank', 'English'

Suppose now that the User's Profile Resolver gives the semantics with the corresponding reference value as is the one declared in the TV-Anytime User's Profile, shown in table 5.

Language	**PVal**
english	80

Table 5: Semantics based on the TV-Anytime User's Profile for the words 'Frank', 'English'

The results in this case, based on the above, are shown in table 6.

	Creator	**Role**	**Language**
1	Frank / English	actor/actor	-
2	Franke / English	director/actor	-
3	Frank	actor	English

Table 6: Results after resolving the ambiguity of the sub-phrase '…with Frank and English'

An example for the case where there is a match between the stop list and the words with the ambiguities but there is not 'and' or an 'or' between the words with the ambiguities would be the sub-phrase **'… with Frank in English'**. Suppose that the TV-Anytime Semantics Resolver gives the semantics shown in table 7.

Creator	**Role**	**Language**
Frank English	actor	english
Holger Franke	director	
Frank… 66 results	various	

Table 7: Semantics from the TV-Anytime Semantics Resolver for the words 'Frank', 'English'

Suppose now that the User’s Profile Resolver gives the semantics with the corresponding reference value as this is declared in the TV-Anytime User’s Profile, shown in table 8.

Language	PVal
English	80

Table 8: Semantics based on the TV-Anytime User’s Profile for the words ‘Frank’, ‘English’

The results in this case, based on the above, are shown in table 9.

	Creator	Role	Language
1	Frank	actor / language	English
2	Frank / English	actor/actor	-
3	Franke / English	director/actor	-

Table 9: Results after resolving the ambiguity of the sub-phrase ‘…with Frank in English’

4 System Architecture

The complete architecture of the system is presented in figure 5. This architecture follows a multi-tier approach and consists of three tiers. The lowest tier handles the metadata management. The middleware tier includes all the logic for interfacing the system with the outside world. The application tier enables the exchange of information between the server and heterogeneous clients through different communication links.

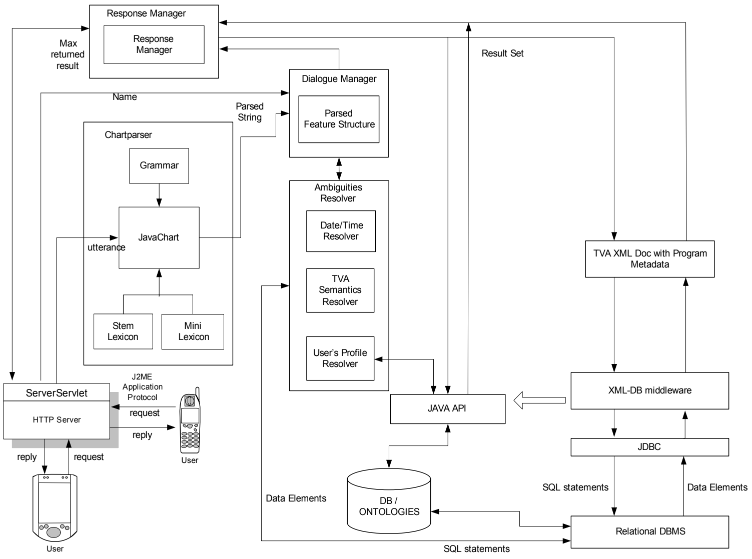

Figure 5: The Natural Language System Architecture

The user uses a wireless device (mobile phone, PDA) and inserts an utterance that is forwarded to the **ChartParser module**. The ChartParser module consists of the JavaChart parser [10] that creates a feature structure with the information from the user's input. There exist two lexicons, the stem lexicon, which contains the stems of the words used in this language, and the words that help to give the right values to the TV-Anytime categories, and the mini lexicon, which contains the endings of the words. Finally, there is the grammar that follows a unified-based formalism.

The **Dialogue Manager** acts as the core module of the system and is responsible for communicating with all the other system modules. It takes as input a list of feature structures from the chart parser and the user's data from the application that he/she is using. It checks for specific properties in order to eliminate the list of the feature structures and, in the case there are any ambiguities, it passes to the Ambiguities Resolver module the list of the words with the ambiguities. Finally, it creates a structure with information about the action, the target and the TV-Anytime XML document from the user's input.

The **Relational Database** of the system contains the TV-Anytime Metadata information, as well as a number of **ontologies** (alternative classification schemes to represent genre hierarchies and creator roles as well as more advanced semantic descriptions following the Semantic Part of the MPEG-7 MDS), with information concerning digital TV. The **Relational DBMS** manages the transactions, utilizes a **Java API** (implemented for the extraction of the functionality for filtering, retrieval and summarization) and cooperates with the **XML-DB middleware**. The **XML-DB middleware** is a set of software components responsible for the management of the TV-Anytime XML documents and the correspondence of the TV-Anytime Metadata XML schema with the underlying relational schema.

The **Ambiguities Resolver** module consists of three modules that are responsible for the resolution of different kinds of ambiguities. The Date/Time Resolver is the component that converts the temporal phrases in a TV-Anytime compliant form. The TVA Semantics Resolver communicates with the relational DBMS and is responsible to attach TV-Anytime semantics to the words with the ambiguities. The User's Profile Resolver filters the list of the words from any existing user's profile and returns a FilteringAndSearchPreferences XML document with values from the corresponding TV-Anytime categories. Finally, it passes this document to the Response Manager module.

The **Response Manager** module interacts with the system's database, by providing it the structured information, executes the appropriate functions, retrieves the results and classifies them accordingly. Then, it creates a message and adds it to any existing result list.

5 Implementation Environment

For the system's server side, the implementation platform consists of the MySQL Platform [13]. MySQL is an open source relational database management system, which is reliable and easy to use. The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces. The implementation of the server was based on Java 2 and the class files were compiled using JDK1.4.1. [14]. The parsing of the user's utterance was made by the use of the JavaChart parser [10], a chart parser written in Java. For the implementation of the communication between the server and the client, we have exploited the JAVA Serlvet technology in the server side by developing a servlet that acts as the interface between the user client and the database server or the PDR interface. This servlet was locally deployed for testing on an Apache Tomcat v4.0.1 server and the class files were compiled using JDK1.4.1.

Two cases are considered related to the wireless device used on the system's client side. The first one is the application that runs on any Java-enabled mobile phone device and the second is the application that runs on a PDA device.

For the client (mobile device) the implementation platform consists of the Java 2 Platform, Micro Edition (J2ME). The platform addresses the large, rapidly growing consumer space, from pagers till TV set-top boxes. The Connected Limited Device Configuration (CLDC) has been used for the limited capabilities of the mobile phone. The Mobile Information Device profile (MIDP) is an architecture and a set of Java libraries that create an open, third party application development environment for small, resource-constrained, devices MIDlets. By sub classing the MIDlet class, an interface between the application and the application management software on the device is defined.

For the second implementation of the client side we used JEODE Java Runtime Environment [15] for the PDA device client. The JEODE runtime environment supports the CDC/Foundation Profile and the Personal Profile J2ME specifications that support implementations in PersonalJava and EmbeddedJava. The PersonalJava platform is a Java platform optimized for the requirements and constraints of wireless devices. The PersonalJava application-programming interface (API) is a collection of packages, classes and methods defined by a high-level specification.

For both cases of the remote access application, the client establishes an http connection with the server.

6 System Flow

In this section we describe an example of a user's utterance and the way that this input flows inside the modules of the system.

We consider the user's utterance

- 'Are there any movies with Frankie Muniz after 5 pm'

The ChartParser module obtains the utterance, as a string, from the wireless device. The output from the parser is the feature structure presented below:

```
[[ properties: [ keyword:  [ 1: Muniz]
                           [ 0: Frankie]]
               [ genre1: movies]]
 [ temporal:   [ timeind:  pm]
               [ after: [ time : 5 ]]
 [ markers:    [ act: ret ]]]
```

The feature structure has a type keyword with values ‘Frankie’ and ‘Muniz’. For that the Dialogue Manager module first creates the vector of the keywords that have ambiguities (Frankie, Muniz) and then parses these words from the stop list and checks if there is an ‘and’ or an ‘or’ among them. The stop list does not remove any of the words and the Dialogue Manager passes the vector with the words to the Ambiguities Resolver module. The TV-Anytime Semantics Resolver (after communicating with the metadata management system) returns a vector from vectors with the semantics for every word:

[[ACTOR, Frankie Muniz]]

The Dialogue Manager module collects the information provided by the Ambiguities Resolver module, plus the personal data (name, password) from the user’s input and creates a structure of the following form:

String action = retrieval	**String target** = null	XML doc

The XML document produced for the utterance applied is presented in figure 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<tva:TVAMain xmlns:tva="http://www.tv-anytime.org/2001/08/metadata"
xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001">
<tva:UserDescription>
   <tva:UserPreferences>
      <mpeg7:UserIdentifier>
            <mpeg7:Name>__temp_faspmatching_</mpeg7:Name>
      </mpeg7:UserIdentifier>
      <mpeg7:FilteringAndSearchPreferences preferenceValue="100">
            <mpeg7:CreationPreferences preferenceValue="100">
            <mpeg7:Creator preferenceValue="100">
            <mpeg7:Role href="urn:mpeg:MPEG7RoleCS:ACTOR" />
            <mpeg7:Agent  xsi:type="mpeg7:PersonType">
            <mpeg7:Name>
                  <mpeg7:GivenName>FrankieMuniz</mpeg7:GivenName>
            </mpeg7:Name>
        </mpeg7:Agent>
      </mpeg7:Creator>
      <mpeg7:DatePeriod preferenceValue="100">
            <mpeg7:TimePoint>2003-08-21T17:00:00</mpeg7:TimePoint>
            <mpeg7:Duration></mpeg7:Duration>
      </mpeg7:DatePeriod>
   </mpeg7:CreationPreferences>
<mpeg7:ClassificationPreferences preferenceValue="100">
            <mpeg7:Genre href="urn:mpeg:TVAnytime_v0.1ContentCS:6" preferenceValue="100">
                  <mpeg7:Name>Movies</mpeg7:Name>
                  <mpeg7:Definition>Film, Cinema</mpeg7:Definition>
            </mpeg7:Genre>
       </mpeg7:ClassificationPreferences>
      <mpeg7:SourcePreferences preferenceValue="100" />
      </mpeg7:FilteringAndSearchPreferences>
    </tva:UserPreferences>
```

Figure 6: The XML doc created by the Dialogue Manager module with the user’s input information

In the previous example there is no information provided by the User's Profile Resolver. The Response Manager module first checks the fields 'action' and 'target' of the structure. Since the action is a retrieval with no target, the search should be done in the metadata of the TV-Anytime server. The XML-DB Middleware inserts the TVA XML doc into the relational database with a temporary value in the field Name of the UserIdentifier. The Response Manager module uses a procedure to match the information with the metadata and returns the matching programs with a weight value that represents the relevance between the information and the result. The results, in the form of programs' titles, are presented to the user with a proper message.

The user can set his preferences for the information about the programs metadata that he would like to obtain as a result while interacting with the system. This information can be also enriched with the program's broadcast date and time, broadcast channel and synopsis; something that mainly depends on the resource capabilities of the device that the user uses to communicate with the natural language system.

7 Evaluation

This section presents some preliminary results of the system's evaluation. The evaluation has been based on a user experiment that was performed by ten laboratory users. All the users had previous experience using computers and graphical interfaces. Nevertheless none of them had ever used a natural language system before. There were three main tasks used in the evaluation. The first one was for the users to use the system to define their user profile, the second one to interact with the system by using a set of utterances with no ambiguities and the third one to interact with the system by using a set of 10 utterances with ambiguities.

The functionality provided by the natural language system was also provided by the use of alternative menu-driven user interfaces for wireless devices (mobile phone, PDA). Based on the preliminary results it becomes clear that the end users found the system easier to use with the natural language interface than with the traditional user interfaces. The natural language interface was shown to provide an easy way to specify TV-Anytime structures without complex navigation between screens. For utterances that showed no ambiguities the system proved to fully exploit the structured TV-Anytime model capabilities for filtering in order to retrieve the qualified ranked results.

In the case of the utterances with ambiguities, we have considered 100 interactions in total. All the utterances contained a sub-phrase with no declaration of any TV-Anytime categories. For example, two of the utterances considered were:

- I want comedies with Tom English
- Record movie from Russia with love

In order to evaluate our approach for the use of the existing user profiles in order to rank the possible answers in case of ambiguities we have considered different types of user profiles so that the similarity between the user's preferences in these profiles and the specific input utterances to be near 0,5.

Diagram 1 shows the number of interactions per rank position for the cases that the system has either used the user profile in order to better rank the results or not. When the system uses the user profile to rank the results, we get about 90% of the exact results in the first 20 positions of the ranked resulting lists. This percentage varies according to the result list. In the first 5 positions we get the 65% of the exact results. When the system does not use the user profile to rank the results, we get about 80% of the exact results in the first 20 positions of the ranked resulting lists. In the first 5 positions we get the 40% of the exact results.

Diagram 2 shows the number of interactions in the top percentage of the total number of results. When the system uses the user profile to rank the results we get about 90% of the exact results in the 40% of the total number of results. In the case that the system does not use the user profile we get about 90% of the exact results in the 70% of the total number of results.

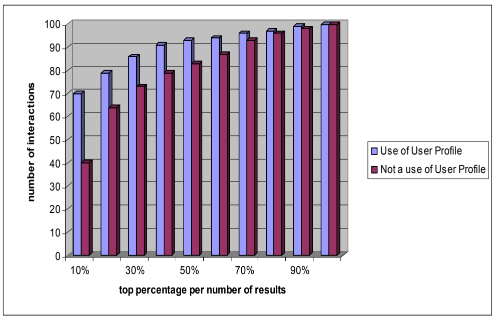

Diagram 1: Number of interactions per rank position

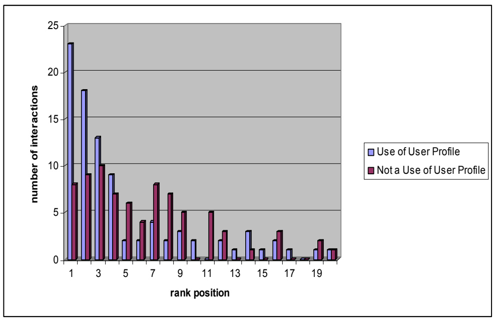

Diagram 2: Number of interactions per top percentage of the total number of results

More systematic evaluation work is still under way.

8 Summary

In this paper we described the design and the implementation of a natural language model in digital TV and mobile environments that allow the user, by using a variety of devices to communicate with the Personal Digital Recorder of his TV set, in order to perform a variety of functions

The model was developed to be compatible with the TV-Anytime specifications and manages the information of the content metadata categories for TV programs and the user's profile metadata. It is expandable to future additions in the TV-Anytime metadata specifications.

In order to satisfy the proposed functionality a dialogue model was developed that contains: Introduction phrases, to define the functionality, Search phrases, to define the TV-Anytime information, Target phrases, to define where each of the functions is targeting, Temporal phrases, to define phrases about date and time and Summary phrases, to define summaries with audio/visual content.

In the proposed natural language model, an utterance can contain one or more entities. These entities distinguish the information, which describes TV-Anytime content metadata information from the information that concerns the functions for its management.

In order to resolve possible ambiguities contained in the user's utterance, the system firstly checks for specific words in the utterance, then searches existing ontologies and attaches semantics to every word that appears with ambiguity and finally checks the TV-Anytime user's profile to attach a weight value to the search results. The algorithm checks for the best cross-correlations and unifies the results by assigning the regular weight values to the results.

The implementation of the natural language model runs on a mobile phone and a PDA, and preliminary evaluation studies have shown it to be a good tool for these environments, better than traditional PC interfaces. Our current research aims to show that natural language (and speech) interfaces are appropriate interface styles for accessing audiovisual content, stored in home information servers, from mobile devices.

9 Reference:

1. The site of the TV-Anytime Forum, http://www.tv-anytime.org
2. ETSI TS 102 822-3-1 V1.1.1 (2003-10) Technical Specification. *Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 3: Metadata;*
3. Kazasis, F.G., Moumoutzis, N., Pappas, N., Karanastasi, A., Christodoulakis, S. (2003). *Designing Ubiquitous Personalized TV-Anytime Services.* In the International Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS), Klagenfurt/Velden, Austria.
4. Johansson, P., Degerstedt, L., Jönsson, A. (2002). *Iterative Development of an Information-Providing Dialogue System.* In Proceedings of the 7th Workshop on User Interfaces for All. Chantilly, France.
5. Ibrahim, A., Johansson, P. (2002). *Multimodal Dialogue Systems: A Case Study for Interactive TV.* In Proceedings of the 7th Workshop on User Interfaces for All. Chantilly, France.
6. Ibrahim, A., Johansson, P. (2002). *Multimodal Dialogue Systems for Interactive TV Applications.* In Proceedings of 4th IEEE International Conference on Multimodal Interfaces, Pittsburgh, USA. pp. 117-222.
7. Jurafsky, D., Martin, J.H. (2000). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* New Jersey: Prentice Hall.
8. Multimodal Interaction for Information Appliances (MIINA Project), http://www.ida.liu.se/~nlplab/miina/
9. NOKIA – Talkative TV-Guide, Program Guide Information System, http://www.nokia.com/nokia/0,,27727,00.html
10. Degerstedt, L. (2002). *JavaChart User Manual*, http://nlpfarm.sourceforge.net/javachart/
11. IST Project UP-TV: "Ubiquitous Personalized Interactive Multimedia TV Systems and Services", http://www.up-tv.de
12. Maier E. (1997). *Clarification dialogues in VERBMOBIL.* In Proceedings of EuroSpeech-97, pp. 1891-1894, Rhodes, 1997
13. MySQL: The World's Most Popular Open Source Database, http://www.mysql.com/
14. Java Technology, http://java.sun.com/
15. The Java 2 Platform, Micro Edition (J2ME™), http://java.sun.com/j2me/