A novel approach for data fusion and dialog management in user-adapted multimodal dialog systems

David Griol, Jesús García-Herrero, José Manuel Molina
Applied Artificial Intelligence Group
Computer Science Department
Carlos III University of Madrid - Spain
Email: {david.griol,jesus.garciaherrero,josemanuel.molina}@uc3m.es

***Abstract*—Multimodal dialog systems have demonstrated a high potential for more flexible, usable and natural human-computer interaction. These improvements are highly dependent on the fusion and dialog management processes, which respectively integrates and interprets multimedia multimodal information and decides the next system response for the current dialog state. In this paper we propose to carry out the multimodal fusion and dialog management processes at the dialog level in a single step. To do this, we describe an approach based on a statistical model that takes user's intention into account, generates a single representation obtained from the different input modalities and their confidence scores, and selects the next system action based on this representation. The paper also describes the practical application of the proposed approach to develop a multimodal dialog system providing travel and tourist information.**

I. Introduction

Speech and natural language technologies allow users to communicate in a flexible and efficient manner, making possible to access applications in which traditional input interfaces cannot be used (e.g. in-car applications, access for disabled persons, etc). Also speech-based interfaces work seamlessly with small devices (e.g., smarthphones and tablets PCs) and allow users to easily invoke local applications or access remote information. For this reason, multimodal dialog systems [1] are becoming a strong alternative to traditional graphical interfaces which might not be appropriate for all users and/or applications.

There are several approaches to make contents available using speech. Some systems add a vocal interface to an existing web browser [2]. Others are focused on specific tasks, as e-commerce [3], chat functionalities [4], database access [5], health services access [6], surveys [7], recommendations systems [8], etc. Finally, the solution could be restricted to access information of a limited domain, like in [9], where the dialog system works for selected on-line resources. From the opposite point of view, some traditional Information Retrieval and Question Answering systems have been extended with a vocal interface, [10].

However, the adaptation capabilities of speech interfaces for mobile devices are frequently restricted to static choices [11], [12]. For example, users have diverse ways of communication. Novice users and experienced users may want the interface to behave completely differently, such as maintaining more guided vs. more flexible dialogs. Processing context is not only useful to adapt the systems' behavior, but also to cope with the ambiguities derived from the use of natural language [13]. For instance, context information can be used to resolve anaphoric references depending on the context of the dialog or the user location. The performance of a dialog system also depends highly on the environmental conditions, such for example whether there are people speaking near the system or the noise generated by other devices.

In this paper, we propose a framework to develop context-aware multimodal dialog systems for mobile devices. Our framework allows to dynamically incorporate user specific requirements and preferences as well as characteristics about the interaction environment, in order to improve and personalize web information and services. The proposed framework is mainly focused on three specific processes carried out by dialog system: context adaptation, fusion of input information sources, and dialog management.

Research in techniques for user modeling has a long history within the fields of language processing and speech technologies. According to Zukerman and Litman [14], very early examples of user modeling in these fields are dominated by knowledge-based formalisms and various types of logic aimed at modeling the complex beliefs and intentions of agents [15], [16], [17]. In more recent years, dialog systems have tended to focus on cooperative, task-oriented rather than conversational forms of dialog, so that user models are now typically less complex. It is possible to classify the different approaches with regard to the level of abstraction at which they model dialog. This can be either at the acoustic level, the word level or the intention-level.

Intention-level models are particularly useful to generate a compact representation of human-computer interaction. Intentions cannot be observed, but they can be described using the speech-act and dialog-act theories [18], [19]. Two main approaches can be distinguished to the creation of user intention models: rule-based and data or corpus-based. In a rule-based user model, different rules determine the behavior of the system [20], [21]. In this approach the researcher has complete control over the design of the evaluation study. However, these proposals are usually designed ad-hoc for their specific domain using models and standards in which developers must specify

each step to be followed by the user model. This way, the adaptation of the hand-crafted designed models to new tasks is a time-consuming process that implies a considerable effort.

Corpus-based approaches use probabilistic methods to generate the user input, with the advantage that this uncertainty can better reflect the unexpected behaviors of users interacting with the system. Statistical models of user intention have been suggested as the solution to the lack of the data that is required for training and evaluating dialog strategies. Using this approach, the dialog system can explore the space of possible dialog situations and learn enhanced strategies [15]. As will be described in Section II-B, our proposed user intention simulation technique is based on a classification process that considers the complete dialog history by incorporating several knowledge sources, combining statistical and heuristic information to enhance the dialog model.

We propose to complement user-adaptation by means of the acquisition of external context using sensors currently supported by Android mobile devices [22]. Most Android-powered devices provide built-in sensors that measure motion, orientation, and various environmental conditions. These sensors are capable of providing raw data to monitor three-dimensional device movement or positioning, and changes in the ambient environment near a device. The Android platform supports three broad categories of sensors. Motion sensors (e.g., accelerometers, gravity sensors, gyroscopes, and rotational vector sensors) measure acceleration forces and rotational forces along three axes. Environmental sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers. Position sensors measure the physical position of a device (e.g., orientation sensors and magnetometers).

Finally, dialog management has the main goal of selecting the next action of the system [23], [24], [25], interpreting the incoming semantic representation of the user input in the context of the dialog. In addition, it resolves ellipsis and anaphora, evaluates the relevance and completeness of user requests, identifies and recovers from recognition and understanding errors, retrieves information from data repositories, and decides about the next system's response.

Automating dialog management is useful for developing, deploying and re-deploying applications and also reducing the time-consuming process of hand-crafted design. In fact, the application of machine learning approaches to dialog management strategy design is a rapidly growing research area. Machine-learning approaches to dialog management attempt to learn optimal strategies from corpora of real human-computer dialog data using automated "trial-and-error" methods instead of relying on empirical design principles [26]. The main trend in this area is an increased use of data for automatically improving the performance of the system and develop systems that exhibit more robust performance, improved portability, better scalability and easier adaptation to other tasks.

In this paper, we propose to merge the multimodal data fusion and dialog management processes by means of a statistical methodology that considers the set of input information sources (spoken interaction, external context acquisition, and user intention modeling), uses a data structure to store the values for the different input information sources received by the dialog manager along the dialog history, and selects the next system response by means of a classification process that takes this data structure as input.

The remainder of the paper is as follows. Section II presents our approach for developing user-adapted multimodal dialog systems. Section III describes the application of our approach to develop a practical system providing travel and tourist information. Section IV presents the results of a preliminary evaluation of this practical dialog system. Finally, Section V presents the conclusions and suggests some future work guidelines.

II. Our proposal to develop user-adapted multimodal dialog systems

Given the number of operations that must be carried out by a dialog system, the scheme used for the development of these systems usually includes several generic modules that deal with multiple knowledge sources and that must cooperate to satisfy the user's requirements. With this premise, a dialog system can be described in terms of the following modules. The *Automatic Speech Recognition module* (ASR) transforms the user utterance into the most probable sequence of words. The *Natural Language Understanding module* (NLU) provides a semantic representation of the meaning of the sequence of words generated by the ASR module. The *Dialog Manager* determines the next action to be taken by the system following a dialog strategy. The *Web Query Manager* receives requests for web services, processes the information, and returns the result to the dialog manager. The *Natural Language Generator module* (NLG) receives a formal representation of the system action and generates a user response that can include multimodal information (video, data tables, images, gestures, etc.), which it is managed by the *Visual Information Generation* module. Finally, a *Text to Speech Synthesizer* (TTS) generates the audio signal transmitted to the user.

As explained in the introduction section, in our contribution, we want also to model the context of the interaction as an additional valuable information source to be considered in the fusion process. We propose the acquisition of external context by means of the use of sensors currently supported by Android devices. Regarding internal context, our proposal is based on the traditional view of the dialog act theory, in which communicative acts are defined as intentions or goals. Our technique is based on a statistical model to predict user's intention during the dialog, which is automatically learned from a dialog corpus. Finally, the fusion of input data and the dialog management processes are merged by means of a statistical methodology that considers the complete history of the dialog. The complete architecture of the user-adapted dialog systems integrating our proposal is shown in Figure 1. The following subsections describes these main components of our proposal.

A. Acquiring external context

As explained in the introduction section, in our contribution we want to model the context of the interaction as an additional valuable information source to be considered in the fusion and dialog management processes.

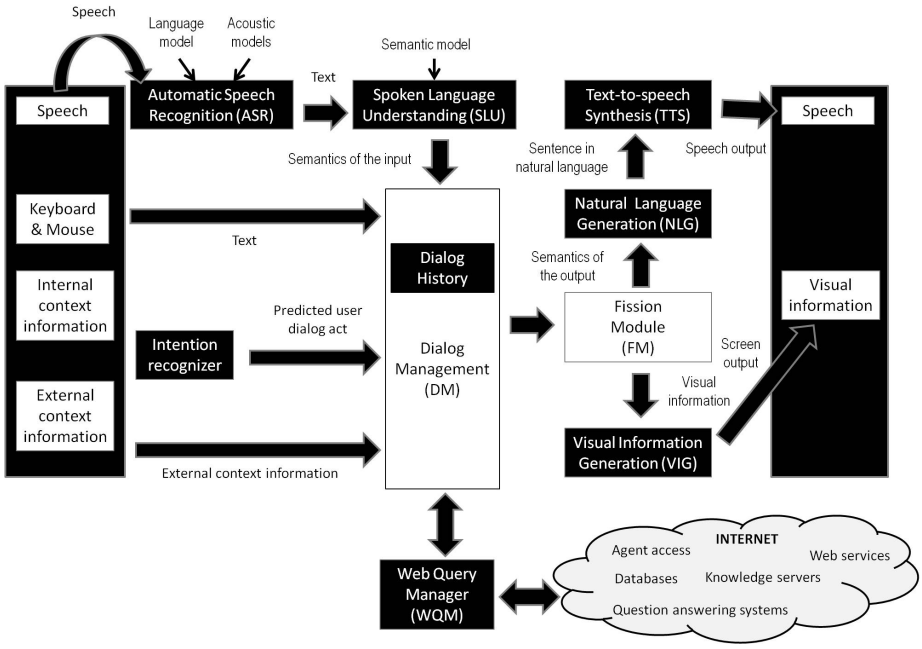

Fig. 1. Architecture of a multimodal dialog system integrating the proposed framework for user-adaptation

We propose the acquisition of external context by means of the use of sensors currently supported by Android devices. Android allows applications to access location services using the classes in the *android.location* package. The central component of the location framework is the *LocationManager* system service, also the Google Maps Android API permits to add maps to the application, which are based on Google Maps data. This API automatically handles access to Google Maps servers, data downloading, map display, and touch gestures on the map. The API can also be used to add markers, polygons and overlays, and to change the user's view of a particular map area. To integrate this API into an application, is it required to install the Google Play services libraries.

Most Android-powered devices have built-in sensors that measure motion, orientation, and various environmental conditions. These sensors are capable of providing raw data with high precision and accuracy, and are useful to monitor three-dimensional device movement or positioning, or monitor changes in the ambient environment near a device. The Android platform supports three main categories of sensors. Motion sensors measure acceleration forces and rotational forces along three axes. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors. Environmental sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers. Finally, position sensors measure the physical position of a device. This category includes orientation sensors and magnetometers.

The Android sensor framework (*android.hardware* package) allows to access these sensors and acquire raw sensor data. Some of these sensors are hardware-based and some are software-based. Hardware-based derive their data by directly measuring specific environmental properties, such as acceleration, geomagnetic field strength, or angular change. Software-based sensors derive their data from one or more of the hardware-based sensors (e.g., linear acceleration and gravity sensors).

Android also provides several sensors to monitor the motion of a device. Two of these sensors are always hardware-based (the accelerometer and gyroscope), and three of these sensors can be either hardware-based or software-based (the gravity, linear acceleration, and rotation vector sensors). Motion sensors are useful for monitoring device movement, such as tilt, shake, rotation, or swing. All of the motion sensors return multi-dimensional arrays of sensor values for each SensorEvent. Two additional sensors allow to determine the position of a device: the geomagnetic field sensor and the orientation sensor. The Android platform also provides a sensor to determine how close the face of a device is to an object (known as the proximity sensor). The geomagnetic field sensor and the proximity sensor are hardware-based. The orientation sensor is software-based and derives its data from the accelerometer and the geomagnetic field sensor.

Finally, four sensors allow monitoring various environmental properties: relative ambient humidity, light, ambient pressure, and ambient temperature near an Android-powered device. All four environment sensors are hardware-based and are available only if a device manufacturer has built them into a device. With the exception of the light sensor, which most device manufacturers use to control screen brightness, environment sensors are not always available on devices. Unlike most motion sensors and position sensors, environment sensors return a single sensor value for each data event.

B. The user intention recognizer

The methodology that we have developed for modeling the user intention extends our previous work in statistical models

for dialog management [25]. We define user intention as the predicted next user action to fulfill their objective in the dialog. It is computed taking into account the information provided by the user throughout the dialog history, and the last system turn. The formal description of the proposed model is as follows. Let A_i be the output of the dialog system (the system response) at time i, expressed in terms of dialog acts. Let U_i be the semantic representation of the user intention. We represent a dialog as a sequence of pairs (*system-turn*, *user-turn*)

$$(A_1, U_1), \cdots, (A_i, U_i), \cdots, (A_n, U_n)$$

where A_1 is the greeting turn of the system, and U_n is the last user turn.

We refer to a pair (A_i, U_i) as S_i, the state of the dialog sequence at time i. Given the representation of a dialog as this sequence of pairs, the objective of the user intention recognizer at time i is to select an appropriate user response U_i. This selection is a local process for each time i, which takes into account the sequence of dialog states that precede time i and the system answer at time i. If the most likely user intention level U_i is selected at each time i, the selection is made using the following maximization rule:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | S_1, \cdots, S_{i-1}, A_i) \tag{1}$$

where set $\mathcal{U}$ contains all the possible user answers.

As the number of possible sequences of states is very large, we establish a partition in this space (i.e., in the history of the dialog up to time i). Let UR_i be what we call user register at time i. The user register can be defined as a data structure that contains information about concepts and attributes values provided by the user throughout the previous dialog history. The information contained in UR_i is a summary of the information provided by the user up to time i. That is, the semantic interpretation of the user utterances during the dialog and the information that is contained in the user profile.

The user profile is comprised of user's:

- Id and user's name, which he can use to log in to the system.
- Gender.
- Experience, which can be either 0 for novel users (first time the user calls the system) or the number of times the user has interacted with the system.
- Skill level, estimated taking into account the level of expertise, the duration of their previous dialogs, the time that was necessary to access a specific content, and the date of the last interaction with the system. A low, medium, high, or expert level is assigned using these measures.
- Most frequent objective of the user.
- Reference to the location of all the information regarding the previous interactions and the corresponding objective and subjective parameters for the user.

The partition that we establish in this space is based on the assumption that two different sequences of states are equivalent if they lead to the same UR. After applying the above considerations and establishing the equivalence relations in the histories of dialogs, the selection of the best U_i is given by:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | UR_{i-1}, A_i) \tag{2}$$

We propose the use of a classification process to predict the user intention following the previous equation. Specifically, we use a multilayer perceptron (MLP) for the classification, where the input layer received the current situation of the dialog, which is represented by the term (UR_{i-1},A_i). The values of the output layer can be viewed as the a posteriori probability of selecting the different user intention given the current situation of the dialog.

C. Fusion of input modalities and dialog management

As previously described, the objective of fusion in multimodal dialog systems is to process the input information and assign a semantic representation which is eventually sent to the dialog manager. Two main levels of fusion are often used: feature-level fusion, semantic-level fusion. The first one is a method for fusing low-level feature information from parallel input signals within a multimodal architecture. The second one is a method for integrating semantic information derived from parallel input modes in a multimodal architecture.

Semantic-level fusion is usually involved in the dialog manager and needs to consult the knowledge source from the dialog history and data repositories. Three popular semantic fusion techniques are used. Frame-based fusion is a method for integrating semantic information derived from parallel input modes. Unification-based fusion is a logic-based method for integrating partial meaning fragments derived from two input modes into a common meaning representation during multimodal language processing. Hybrid symbolic/statistical fusion is an approach to combine statistical processing techniques with a symbolic unification-based approach (e.g. Members-Teams-Committee (MTC) hierarchical recognition fusion.

The methodology that we propose for the multimodal data fusion and dialog management processes considers the set of input information sources (spoken interaction, external context acquisition, and user intention modeling) by means of a machine-learning technique that extends our proposal for user modeling. In a similar way, we propose the definition of a data structure to store the values for the different input information sources received by the dialog manager along the dialog history. The information stored in this data structure, that we called Interaction Register (IR), is coded in terms of three values, $\{0, 1, 2\}$, for each field according to the following criteria:

- **0**: The value of the specific position of the IR has not been provided by means of any of the input modalities or sources defined as interaction context.
- **1**: The value of the specific position of the IR has been provided with a confidence score that is higher than a given threshold. Confidence scores are provided by different modules that process the information

acquired for each input modality (e.g., the ASR and SLU modules for the spoken utterances).

- **2**: The value of the specific position of the IR has been provided with a confidence score that is lower than the given threshold.

The information in the IR at each time i is thus generated considering the values extracted from the inputs to the dialog manager along the dialog history. Each slot in the IR can be usually completed by means of an input modality or by the use of the external context. If just one value has been received for a specific dialog act, then it is stored at the corresponding slot in the IR using the described codification. Confidences scores provided by the modules processing each input modality are used in case of conflict among the values provided by several modalities for the same slot. Thus, a single input is generated for the dialog manager to consider the next system response. The predicted user dialog act (generated by means of Equation 2) is also incorporated as an additional slot of the IR. After applying the above considerations, the selection of the best system response A_i is given by Equation 3.

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | IR_{i-1}, A_{i-1}) \tag{3}$$

As in our previous work on user modeling, we propose the use of a classification process to determine the next system response given the single input that is provided by the interaction register after the fusion of the input modalities and also considering the previous system response. This way, the current state of the dialog is represented by the term (IR_i, A_{i-1}), where A_{i-1} represents the last system response. The values of the output of the classifier can be viewed as the a posteriori probability of selecting the different system responses given the current situation of the dialog.

III. Practical application

We have applied our context aware methodology to develop and evaluate an adaptive multimodal dialog system for a travel-planning domain. The system provides context-aware information in natural language in Spanish about approaches to a city, flight schedules, weather forecast, car rental, hotel booking, tourist attractions, theater listings, and film showtimes. The information offered to the user is extracted from a web page that users can visually complete to incorporate additional information about a city already present in the system, update this information or add new cities. Different Postgress databases are used to store this information and automatically update the data that is included in the application. In addition, several functionalities are related to dynamic information (e.g., weather forecast, flight schedules) directly obtained from webpages and web services. Thus, our system provides speech access to facilitate travel-planning information that is adapted to each user taking context into account.

Semantic knowledge is modeled in our architecture using the classical frame representation of the meaning of the utterance. We defined eight concepts to represent the different queries that the user can perform (*City-Approaches*, *Flight-Schedules*, *Weather-Forecast*, *Car-Rental*, and *Hotel-Booking*, *Tourist-Attractions*, *Heater-Listings*, and *Film-Show times*). Three task-independent concepts have also been defined for the task (*Affirmation*, *Negation*, and *Not-Understood*). A total of 101 system actions (DAs) were defined taking into account the information that the system provides, requests or confirms.

Using the *City_Approaches* functionality, it is possible to know how to get to a specific city using different means of transport. If specific means are not provided by the user, then the system provides the complete information available for the required city. Users can optionally provide an origin city to try to obtain detailed information taking into account this origin. Context information taken into account to adapt this information includes user's current position, and preferred means of transport and city.

The *Flight_Schedules* functionality provides flight information considering the user's requirements. Users can provide the origin and destination cities, ticket class, departure and/or arrival dates, and departure and/or arrival hours. Using *Weather_Forecast* it is possible to obtain the forecast for the required city and dates (for a maximum of 5 days from the current date). For both functionalities, this information is dynamically extracted from external webpages. Context information taken into account includes user's current location, preferred dates and/or hours, and preferred ticket class.

The *Car_Rental* functionality provides this information taking into account users' requisites including the city, pick-up and drop-off date, car type, name of the company, driver's age, and office. The provided information is dynamically extracted from different webpages. The *Hotel_Booking* functionality provides hotels which fulfill the user's requirements (city, name, category, check-in and check-out dates, number of rooms, and number of people).

The *Tourist-Attractions* functionality provides information about places of interest for a specific city, which is directly extracted from the webpage designed for the application. This information is mainly based on users recommendations that have been incorporated in this webpage. The *Theatre_Listings* and *Film_Showtimes* respectively provide information about theater performances and film showtimes that takes into account the users requirements. These requirements can include the city, name of the theater/cinema, name of the show/film, category, date, and hour. This information is also considered to adapt both functionalities and then provide context-aware information.

An example of the semantic interpretation of a user utterance using the list of described dialog acts described is shown in Figure 2.

The IR defined for the task is a sequence of 57 fields, corresponding to:

- The eight possible queries that users can perform to the system (*City-Approaches*, *Flight-Schedules*, *Weather-Forecast*, *Car-Rental*, and *Hotel-Booking*, *Tourist-Attractions*, *Theater-Listings*, and *Film-Showtimes*).
- A total of 45 possible attributes that users can provide to the system in order to generate a detailed response for the different queries (e.g., *Origin_City*, *Destination_City*, *Country*, *Departure_Date*,

Input sentence:
[SPANISH] *Sí, me gustaría conocer los accesos en coche y los hoteles de cuatro estrellas disponibles en Valencia para mañana.*
[ENGLISH] *Yes, I would like to know how to get to Valencia by car and which four stars hotels are available for tomorrow.*
Semantic interpretation:
(*Affirmation*)
(*City_Approaches*)
City: Valencia
Means_Transport: Car
(*Hotel_Booking*)
City: Valencia
Hotel_Booking: Car
Category: Four Stars
Check_in_Date: Tomorrow

Fig. 2. An example of the labeling of a user turn in the travel-planning system

Departure_Hour, *Arrival_Date*, *Hotel_Name*, *Hotel_Category*, *Check_in_Date*, *Check_out_Date*, *Number_Rooms*, *Number_People*, *Category*, *Film*, *Cinema*, *Show*, *Theater*, etc.).

- Three task-independent concepts that users can provide (*Acceptance*, *Rejection* and *Not-Understood*).
- A reference to the predicted user response provided by the user intention recognizer.

A set of 150 scenarios were manually defined to cover the different queries to the system including different user requirements and profiles. Basic scenarios defined only one objective for the dialog; i.e. the user aims at obtaining information about only one type of the possible queries to the system (e.g., to obtain flight schedules from an origin city to a destination for a specific date). More complex scenarios included more than one objective for the dialog (e.g., to obtain information about how to get to a specific city, as well as car rental and hotel booking information).

IV. Experiments

We have completed a preliminary evaluation of our proposal by developing two multimodal dialog systems for the described task. The *Baseline system* does not integrate our proposed framework for the context-adaptation of the system and the *Context-aware system* includes the required modules for context-adaptation as Figure 1 shows.

A total of 150 dialogs were recorded from interactions of six users employing the Baseline and Context-aware systems. The evaluation was carried out by students and lecturers in our department following the types of scenarios described in the paper in different settings with their own devices. An objective and subjective evaluation were carried out. We considered the following measures for the objective evaluation:

1) Dialog success rate. This is the percentage of successfully completed tasks. In each scenario, the user has to obtain one or several items of information, and the dialog success depends on whether the system provides correct data (according to the aims of the scenario) or incorrect data to the user.
2) Average number of turns per dialog (nT).
3) Confirmation rate. It was computed as the ratio between the number of explicit confirmations turns (nCT) and the number of turns in the dialog (nCT/nT).
4) Average number of corrected errors per dialog (nCE). The average of errors detected and corrected by the dialog manager. We have considered only those which modify the values of the attributes and thus could cause the failure of the dialog. The errors are detected using the confidence scores provided by the ASR and NLU modules. Implicit and explicit confirmations are employed to confirm or require again values detected with low reliability.
5) Average number of uncorrected errors per dialog (nNCE). This is the average of errors not corrected by the dialog manager. Again, only errors that modify the values of the attributes are considered.
6) Error correction rate (%ECR). The percentage of corrected errors, computed as nCE/ (nCE + nNCE).

The results presented in Table I show that both systems could interact correctly with the users in most cases. However, the context-aware system obtained a higher success rate, improving the context-unaware results by 12% absolute. Using the context-aware system, the average number of required turns is also reduced from 15.6 to 8.4. These values are slightly higher for both systems as in some dialogs the real users provided additional information which was not mandatory for the corresponding scenario or asked for additional information not included in the definition of the scenario once its objectives were achieved.

The confirmation and error correction rates were also improved by the context-aware system, given that less information is required to the user, reducing the probability of introducing ASR errors. The main problem detected was related to user inputs misrecognized with a very high ASR confidence, and this erroneous information was forwarded to the dialog manager. However, as the success rate shows, this fact did not have a considerable impact on the system operation.

In addition, we asked the users to complete a questionnaire to assess their subjective opinion about the system performance. The questionnaire had five questions: i) Q1: *How well did the system understand you?*; ii)Q2: *How well did you understand the system messages?*; iii) Q3: *Was it easy for you to get the requested information?*; iv) Q4: *Was the interaction rate adequate?*; v) Q5: *Was it easy for you to correct the system errors?*. The possible answers for each one of the questions were the same: *Never*, *Seldom*, *Sometimes*, *Usually*, and *Always*. All the answers were assigned a numeric value between one and five (in the same order as they appear in the questionnaire). Table II shows the average results of the subjective evaluation.

From the results, it can be observed that both systems are considered to correctly understand the different user queries and obtain a similar evaluation regarding the facility of correcting errors introduced by the ASR module. However, the

	Success Rate	nT	Confirmation Rate	%ECR	nCE	nNCE
Baseline system	82%	15.6	29%	78%	0.82	0.23
Context-Aware system	94%	8.4	26%	87%	0.91	0.14

TABLE I. RESULTS OF THE OBJECTIVE EVALUATION OF THE CONTEXT-AWARE AND CONTEXT-UNAWARE SYSTEMS WITH REAL USERS

	Q1	Q2	Q3	Q4	Q5
Baseline system	4.1	4.8	3.9	3.6	3.2
Context-Aware system	4.3	4.7	4.6	4.5	3.5

TABLE II. RESULTS OF THE SUBJECTIVE EVALUATION OF THE BASELINE AND CONTEXT-AWARE SYSTEMS WITH REAL USERS (0=WORST, 5=BEST EVALUATION)

context-aware system has a higher evaluation rate regarding the facility of obtaining the data required to fulfill the complete set of objectives of the scenario and the suitability of the interaction rate during the dialog.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have described a framework to develop dialog systems that considers information provided by means of several input modalities. We carry out an additional step towards the adaptation of these systems by also modeling the context of the interaction in terms of external and internal context, which in our case is respectively related to the acquisition of external context by means of sensors available in mobile devices and the detection of the user's intention.

Using our framework it is possible to develop multimodal interfaces that optimize interaction management and integrate different sources of information that make it possible for the application to adapt to the user and the context of the interaction. To show the pertinence of our proposal, we have implemented an evaluated an Android application that uses geographical context in order to provide different location services to its users. The results show that the users were satisfied with the interaction with the system, which achieved high performance rates. We are currently using the framework to build applications in other increasingly complex domains implying different web services and web services mashups.

ACKNOWLEDGEMENTS

This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485).

REFERENCES

[1] R. Pieraccini, *The Voice in the Machine: Building Computers That Understand Speech*. MIT Press, 2012.

[2] B. Vesnicer, J. Zibert, S. Dobrisek, N. Pavesic, and F. Mihelic, "A voice-driven web browser for blind people," in *Proc. of Interspeech/ICSLP*, 2003, pp. 1301–1304.

[3] M. Tsai, "The VoiceXML dialog system for the e-commerce ordering service," in *Proc. of CSCWD'05*, 2005, pp. 95–100.

[4] M. Kearns, C. Isbell, S. Singh, D. Litman, and J. Howe, "CobotDS: A Spoken Dialogue System for Chat," in *Proc. of AAAI'02*, 2002, pp. 425–430.

[5] T. Nishimoto, Y. Kobayashi, and Y. Niimi, "Spoken Dialog System for Database Access on Internet," in *Proc. of AAAI'97*, 1997, pp. 95–100.

[6] D. Griol, M. McTear, Z. Callejas, R. López-Cózar, N. Ábalos, and G. Espejo, "A methodology for learning optimal dialog strategies," *LNCS*, vol. 6231, pp. 507–514, 2010.

[7] A. Stent, S. Stenchikova, and M. Marge, "Reinforcement learning of dialogue strategies with hierarchical abstract machines," in *Proc. of SLT'06*, 2006, pp. 210–213.

[8] J. Chai, V. Horvath, N. Nicolov, M. Stys, N. Kambhatla, W. Zadrozny, and P. Melville, "Natural language assistant: A dialog system for online product recommendation," *AI Magazine*, vol. 23, pp. 63–75, 2002.

[9] J. Polifroni, G. Chungand, and S. Seneff, "Towards the Automatic Generation of Mixed-Initiative Dialogue Systems from Web Content," in *Proc. of Eurospeech'03*, 2003, pp. 193–196.

[10] E. Sanchis, D. Buscaldi, S. Grau, L. Hurtado, and D. Griol, "Spoken QA based on a passage retrieval engine," in *Proc. of SLT'06*, 2006, pp. 62–65.

[11] S. Whittaker, "Interaction design: what we know and what we need to know," *Interactions*, vol. 20, no. 4, pp. 38–42, 2013.

[12] G. Niklfeld, R. Finan, and M. Pucher, "Architecture for adaptive multi-modal dialog systems based on voiceXML," in *Proc. of Interspeech'01*, 2003, pp. 2341–2344.

[13] S. Seneff, M. Adler, J. Glass, B. Sherry, T. Hazen, C. Wang, and T. Wu, "Exploiting Context Information in Spoken Dialogue Interaction with Mobile Devices," in *Proc. of IMUx'07*, 2007, pp. 1–11.

[14] I. Zukerman and D. Litman, "Natural language processing and user modeling: Synergies and limitations," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 129–158, 2001.

[15] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young, "A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies," *Knowledge Engineering Review*, vol. 21(2), pp. 97–126, 2006.

[16] R. Moore, "Reasoning about knowledge and action," in *Proc. of IJCAI'77*, 1977, pp. 223–227.

[17] P. Bretier and M. D. Sadek, "A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction," in *Proc. of ATAL'96*, 1996, pp. 189–203.

[18] J. Searle, *Speech acts. An essay on the philosophy of language*. Cambridge University Press, 1969.

[19] D. Traum, *Foundations of Rational Agency*. Kluwer, 1999, ch. Speech acts for dialogue agents, pp. 169–201.

[20] G. Chung, "Developing a flexible spoken dialog system using simulation," in *Proc. of ACL'04*, 2004, pp. 63–70.

[21] R. López-Cózar, A. de la Torre, J. Segura, and A. Rubio, "Assessment of dialogue systems by means of a new simulation technique," *Speech Communication*, vol. 40, pp. 387–407, 2003.

[22] M. McTear and Z. Callejas, *Voice Application Development for Android*. Packt Publishing, 2013.

[23] D. Traum and S. Larsson, *The Information State Approach to Dialogue Management*. Kluwer, 2003, ch. Current and New Directions in Discourse and Dialogue, pp. 325–353.

[24] J. Williams and S. Young, "Partially Observable Markov Decision Processes for Spoken Dialog Systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, 2007.

[25] D. Griol, L. Hurtado, E. Segarra, and E. Sanchis, "A statistical Approach to Spoken Dialog Systems Design and Evaluation," *Speech Communication*, vol. 50, no. 8-9, pp. 666–682, 2008.

[26] S. Young, "The Statistical Approach to the Design of Spoken Dialogue Systems," Cambridge University Engineering Department, Tech. Rep., 2002.