Chapter 10
Enhancement of Conversational Agents by Means of Multimodal Interaction

Ramón López-Cózar
University of Granada, Spain

Zoraida Callejas
University of Granada, Spain

Gonzalo Espejo
University of Granada, Spain

David Griol
Carlos III University of Madrid, Spain

ABSTRACT

The main objective of multimodal conversational agents is to provide a more engaged and participative communication by allowing users to employ more than one input methodologies and providing output channels that are different to exclusively using voice. This chapter presents a detailed study on the benefits, disadvantages, and implications of incorporating multimodal interaction in conversational agents. Initially, it focuses on implementation techniques. Next, it explains the fusion and fission of multimodal information and focuses on the core module of these agents: the dialogue manager. Later on, the chapter addresses architectures, tools to develop some typical components of the agents, and evaluation methodologies. As a case of study, it describes the multimodal conversational agent in which we are working at the moment to provide assistance to professors and students in some of their daily activities in an academic centre, for example, a University's Faculty.

DOI: 10.4018/978-1-60960-617-6.ch010

1. INTRODUCTION

Conversational agents can be defined as computer programs designed to interact with users *similarly* as a human being would do, using more or less interaction modalities depending on their complexity (McTear, 2004; López-Cózar & Araki, 2005). These agents are employed for a number of applications, including tutoring (Forbes-Riley & Litman, 2011; Graesser et al., 2001; Johnson & Valente, 2008), entertainment (Ibrahim & Johansson, 2002), command and control (Stent et al., 1999), healthcare (Beveridge & Fox, 2006), call routing (Paek & Horvitz, 2004) and retrieval of information about a variety of services, for example, weather forecasts (Maragoudakis, 2007), apartment rental (Cassell et al., 1999) and travels (Huang et al., 1999).

The implementation of the agents is a complex task in which a number of technologies take part, including signal processing, phonetics, linguistics, natural language processing, affective computing, graphics and interface design, animation techniques, telecommunications, sociology and psychology. The complexity is usually addressed by diving the implementation into simpler problems, each associated with an agent's module that carries out specific functions, for example, automatic speech recognition (ASR), spoken language understanding (SLU), dialogue management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS).

ASR is the process of obtaining a sentence (text string) from a voice signal (Rabiner & Juang, 1993). It is a very complex task given the diversity of factors that can affect the input, basically concerned with the speaker, the interaction context and the transmission channel. Different applications demand different complexity of the speech recognizer. Cole et al. (1997) identified eight parameters that allow an optimal tailoring of the speech recognizer: speech mode, speech style, dependency, vocabulary, language model, perplexity, signal-to-noise ratio (SNR) and transduction. Nowadays general-purpose speech recognition systems are usually based on Hidden Markov Models (HMMs).

SLU is the process of extracting the semantics from a text string (Minker, 1998). It generally involves employing morphological, lexical, syntactical, semantic, discourse and pragmatical knowledge. In a first stage lexical and morphological knowledge allow dividing the words in their constituents distinguishing lexemes and morphemes. Syntactic analysis yields a hierarchical structure of the sentences. Semantic analysis extracts the meaning of a complex syntactic structure from the meaning of its constituents. There are currently two major approaches to tackle the problem of spoken language understanding: rule-based (Mairesse et al., 2009) and statistical (Meza-Ruiz et al., 2008), including some hybrid methods (Liu et al., 2006).

The DM is responsible of deciding the next action to be carried out by the agent. One possible action is to initiate a database query to provide information to the user, for example, available flights connecting two cities. Another possible action is requesting additional data from the user necessary to make the database query, for example, date for a travel. A third typical action is confirming data obtained from the user, for example, departure and arrival cities. This last action is very important given the current limitations of state-of-the-art ASR.

Conversational agents can be divided into two types depending on the interaction modalities available: *spoken* and *multimodal*. The former type allows just speech as the interaction modality (McTear, 2004). Typically, these agents are used to provide telephone-based information, and are comprised of the five main technologies mentioned above, i.e., ASR, SLU, DM, NLG and TTS.

Some of these agents support multilingual interaction, thus enabling the same service for users who speak different languages (Glass et al., 1995). Although agents that process only speech are usable in many cases and for many

application domains, they are very sensitive to the limitations of the speech recogniser. Even though remarkable advances have been made in the last years, state-of-the-art ASR is not mature enough to enable error-free interaction in real world conditions, i.e. regardless of a diversity of factors that degrade its performance, for example, acoustic conditions, user types, accents, speaking styles and vocabulary size.

Another problem is that the interaction can be adapted only partially to different environments and users, given that there is just one interaction modality available (speech). Because of these reasons, among others, many users do not feel comfortable using these agents and reject using them.

After this brief introduction, the reminder of the chapter is organised as follows. Section 2 discusses benefits and disadvantages of multimodal interaction for conversational agents. Section 3 addresses techniques to implement this type of agent, including Wizard of Oz, system-in-the-loop as well as fusion and fission of multimodal information. The section also describes models for implementing the dialogue management (finite states, frames, models based on Artificial Intelligence, statistical approaches and VoiceXML) and addresses methods to implement Embodied Conversational Agents: FACS, MPEG-4 and XML-based languages.

Section 4 focuses on architectures, addressing Galaxy Communicator, Open Agent Architecture (OAA), Blackboard, R-Flow and others. Section 5 describes tools for implementing automatic speech recognition, spoken language understanding, dialogue management, natural language generation, speech synthesis and embodied conversational agents.

Evaluation methodologies are discussed in Section 6, where we address general evaluation frameworks, types and measures, as well as a number of evaluation techniques (PARADISE, PROMISE, CAS and Wizard of Oz). Section 7 describes our latest work in the development a multimodal conversational agent, termed HADA-DS, the goal of which is to provide assistance to professors and students in some of their daily activities in an academic centre, for example, a University's Faculty. Finally, Section 8 presents the conclusions and outlines possibilities for future research directions.

2. MULTIMODAL CONVERSATIONAL AGENTS

Multimodal conversational agents are much more complex than spoken conversation agents. They are based on the fact that human-to-human communication relies on the use of several communication channels to transmit and receive information from the conversation partner, e.g., speech, lip movements, body gestures and gazes (López-Cózar & Araki, 2005).

Humans use all these information channels simultaneously and unconsciously, which enables them to obtain a great performance in understanding messages even in the presence of noise. Imitating this human procedure, multimodal conversational agents can use several input modalities to obtain data from the user, and a number of output modalities to influence several senses of the user simultaneously. This fact poses a number of advantages for the interaction but also some drawbacks, which are discussed in the next section.

2.1 Benefits of Multimodal Interaction

Taking into account the modalities available, the interaction can be carried out employing devices such as microphone, keyboard, mouse, camera, touch-sensitive screen, loudspeaker, display, data glove or haptic hardware (Wahlster, 2006). A benefit of having this wide range of devices available is that the user can select the most appropriate devices considering environmental conditions (e.g. in terms of noise) as well as his

preferences or needs. For example, handicapped users many not be able to use some modality but yet use another.

The user can also select the interaction modalities considering the type of task to be carried out. For example, while driving a car it is safer to use a modality that allows having the hands and eyes free, for example, speech. On the contrary, in a place requiring silence (e.g. a library) or when providing personal data to the agent, speech may not be the best option. Taking into account the adaptation facilities, there are in the literature agents specifically designed for mobile applications, in which proper adaptation to different acoustic conditions is critical (Johnston et al., 2002; Reithinger & Sonntag, 2005).

Another advantage is that the use of several input modalities in parallel for the input to the agent allows compensating to some extent their respective. For example, speech can be employed to reference objects not displayed on screen, whereas graphics can be used to show on screen the effects of performing actions on specific objects (Kuppevelt & Dybkjaer, 2005).

Many multimodal agents adopt a graphical human-like appearance in order to provide a more natural and friendly interaction to the user. Depending on the portion of body shown on screen, they are usually called Talking Heads or Embodied Conversational Agents (ECAs). These characters provide auditory and visual feedback, which is particularly useful when the interaction takes place in noisy environments. Their complexity varies significantly in terms of sophistication and complexity, from simple cartoon-like to complex animated human faces. These characters are connected to the modules of the conversational agent that generate information by means of the output modalities, for example, speech synthesis, lip movements, facial gestures, and video or images on the display.

We can also find in the literature multimodal conversational agents developed for new computer-based paradigms such as ubiquitous computing, pervasive computing or ambient intelligence (Malaka et al., 2004).

2.2 Disadvantages of Multimodal Interaction

In despite of the advantages, enabling multimodal interaction for conversational agents has some drawbacks. For example, some researchers suggest that these agents may impose a greater cognitive load on the user. In fact, there are studies in the literature suggesting that the claimed advantages discussed above are sometimes questionable (Walker et al., 1994; Takeuchi & Nagao, 1995).

In terms of the input to the agents, a problem with using several modalities is that they can provoke ambiguity, contradictions or uncertainty. An example of ambiguity occurs when using a pen on a touch-sensitive screen, the gesture made can be interpreted by several recognisers of the system, e.g. the gesture and the handwriting recognisers, which will create their own recognition hypotheses. Hence, it might not be clear whether the user wanted to make a gesture or write something. An example of contradiction occurs, for example, when interacting with an agent developed for pedestrian navigation, the user says "scroll map to the west" while he draws an arrow on the screen pointing east (Johnston et al., 2002).

Another drawback of multimodal interaction is in terms of the processing of the information captured from the user by means of several modalities. The modalities can cooperate in different ways, and thus can be either processed independently or combined by a process which is typically called *fusion* (Nigay & Coutaz, 1993). Hence, the agent must decide which information chunks correspond to the same input and which ones to different inputs. Making the decision requires implementing complex functions that may consider a number of factors, such as time intervals of the inputs, complementarity of the information chunks and

contextual information. Using these different factors, the agents may decide, for example, to combine information chunks provided in parallel if they are complementary and overlapped in time. The problem is that sometimes this decision might be made with some degree of uncertainty.

In terms of the generation of the agent's output, there might be also a problem concerned with the selection of the most appropriate interaction modalities for a given response to be provided to the user. For example, some responses might be provided using a single modality whereas others might be provided using several. Again, the designers of the agent must decide when and how several modalities must be employed in order to enhance friendliness, yet ensuring not over incrementing the cognitive load of the user. In addition, multimodality requires more computational power in order to ensure correct synchronisation of the output modalities, for example, synthesised speech, facial expressions and gestures of the ECAs (Wahlster, 2003).

3. IMPLEMENTATION TECHNIQUES

This section firstly describes two implementation techniques typically used for developing conversational agents: Wizard of Oz and System-in-the-loop. Secondly, it addresses two processes called *fusion* and *fission* of multimodal information. Third, it discusses techniques employed to implement the core module of these agents: the dialogue manager. To conclude, it comments briefly three methods employed to implement embodied conversational agents: FACS, MPEG-4 and XML-based languages.

3.1 Wizard of Oz

The Wizard of Oz (WOz) is a technique that uses a human called *Wizard* to play the role of the computer in a human-computer interaction (Fraser & Gibert, 1991; Zapata & Carmona, 2007). The users are made to believe that they interact with a computer but actually they interact with the Wizard. This technique has been used in several fields, including test of the software life cycle (Salber & Coutaz, 1993; Fraikin & Leonhardt, 2002), corpus collection (Steininger et al., 2002; Zhang et al., 2005), and spoken or multimodal conversational agents (Mayfield & Burger, 1999; Batliner et al., 2003; Petrelli et al., 1997).

Salber & Coutaz (1993) discussed some requirements of WOz for multimodal conversation agents. They indicated that a multimodal agent is more complex to simulate than an agent based on speech only, which increases the task complexity and the bandwidth necessary for the simulation. For multimodal interaction the authors suggested to employ a multi-wizard configuration, which requires properly organising the work of several wizards. A platform for multimodal WOz experiments must have a high performance and flexibility, and should include a tool to retrieve and manipulate a posteriori data collected during the experiments.

3.2 System-in-the-Loop

The system-in-the-loop technique is based on the fact that software systems improve cyclically by means of user interactions. For example, the performance of a speech-based conversational agent can be improved by means of analyses of sentences previously uttered by users. If modifications are needed in the design of the system, the technique is employed again to obtain new experimental results. These steps (collection of data and test of system) are repeated until the system designers are satisfied with the performance. Among others, Van de Burgt et al. (1996) used this technique to implement the SCHISMA system. Concretely, the technique was used to collect user utterances and analyse them in order to improve the performance of the system.

3.3 Fusion and Fission of Multimodal Information

The *fusion* of multimodal information is a technique to combine information chunks provided by different input modalities of a conversational agent. The result is a data structure that allows the agent to handle simultaneously different information types. Using this data structure the agent's dialogue manager can decide what to do next. A number of methods have been proposed to represent the combined data. For example, Faure and Julia (1993) employed *Triplets*, which are a syntactic formalism to represent multimodal events in the form: (*verb*, *object*, *location*). The authors found this method very useful to represent speech information combined with deictic information generated by means of gestures.

Nigay & Coutaz (1995) employed *Melting pots* to represent combined information, including timestamps. The authors proposed three criteria for deciding whether to carry out or not the fusion process: complementarily of melting pots, time and context. The pots were combined using either *microtemporal*, *macrotemporal* or *contextual* fusion. The first type combined information chunks produced simultaneously or near in time. The second type combined related information chunks, which were generated either sequentially, in parallel or delayed due to insufficient processing power of the system. The third type combined information chunks considering semantic constraints.

Allen (1995) proposed to use semantic structures called *frames*. The information from each modality was interpreted separately and transformed into frames, the slots of which determined the parameters of the action to be made. Frames contained partial information if some slots were empty. During the fusion the frames were combined, which could fulfil slots. For example, Lemon et al. (2006a) used frames to combine multimodal information in a conversational agent that provided information about hotels, restaurants and bars in a town.

Typed Feature Structures (TFS) have also been used to represent *fusioned* multimodal information. The goal is to employ in the fusion process key aspects regarding three formalisms for information representation: unification-based grammar formalisms, languages for knowledge representation and logic programming (Carpenter, 1992; Emele, 1994). For example, Alexandersson & Becker (2001) used this method in the SmartKom agent to handle speech and gestures to book seats in a cinema.

XML-based languages are other method to represent multimodal information. For example, Wahlster (2001) used an XML-based language called M3L to represent all the information flows between the processing components of the SmartKom agent.

The opposite to the fusion technique is called *fission*. The goal of it is to translate each response of the agent into a set of multimodal actions and coordinate the output across the modalities (Müller et al., 2003). This task is very important for multimodal conversational agents in order to make the information be coherently presented to the user. For example, if an ECA makes a reference to an object shown on the display, the reference and the presentation of the object must be properly synchronised. The reference to the object can be carried out using a variety of modalities, for example, deictic gesture of the ECA or highlighting of the object.

The reference can also be cross-modal, using a spoken message such as *"The image on the left corner of the screen..."*. In this latter case, the reference requires that the modality that makes the reference can access the internal representation of the contents on the display. For example, the SmartKom system (Wahlster, 2006) uses a representation for these contents, which allows the visual objects be part of the discourse representation and thus be referenced using several modalities.

3.4 Dialogue Management

The dialogue management is a process that represents the "intelligence" of the conversational agent. It is implemented by means of an agent's component called *dialogue manager*, which analyses the data provided by the different input modalities and decides the next action of the agent, for example, provide information to the user. A number of models can be found in the literature for the implementation of this component. In this section we discuss four of these: finite states, frames, plans and statistical approaches (Allen et al., 2001; McTear, 2004). We also address an XML-based language called VoiceXML for rapid implementation.

3.4.1 Finite States

Dialogue management strategy using finite states is determined beforehand and usually represented as a network (McTear, 1998). Nodes represent agent prompts and transitions represent paths in the network considering user responses, so that the interaction is fully structured. The main advantage of this approach is its simplicity, facilitating the development of dialogue managers when the task is straightforward, clearly structured, and there is a small number of types of system responses. The main drawback is that this approach is unsuitable to manage complex dialogues due to the lack of flexibility, since users must follow the paths defined for the different states.

3.4.2 Frames

The main objective of using frames for dialogue management is to solve the lack of flexibility of the finite state models. Both methodologies are similar in that they are able to manage tasks based on the filling of a form by requesting data from the user. The main difference is that frame-based model does not require following a predefined order to fulfill the required fields, so that it is possible to use a mixed initiative.

To allow this degree of flexibility, it is necessary to provide the system with three main components: i) a frame that refers to the different concepts and attributes defined for the task; ii) a more complete grammar or language model for the ASR module; iii) an algorithm to control the dialogue and determine the next system action based on the contents of the frame. Additional information can be included in the frame definition, for instance, the use of confidence scores to indicate the data reliability.

Goddeau et al. (1996) present a dialogue manager in the domain of cars using this idea. The defined frame, called E-form (electronic form), includes information about the user preferences. These forms are used for dialogue management in the Bell Labs Communicator system (Potamianos et al., 2003), JUPITER (Zue et al., 2000), ARISE (Den et al., 1999), WITAS (Doherty et al., 1998), COMIC (Catizone et al., 2003), to mention a few examples.

3.4.3 Models Based on Artificial Intelligence

We can find in the literature two models based on principles of Artificial Intelligence: *plans* and *agents*. The former takes into account that people plan actions in order to achieve specific goals. Therefore, a dialogue manager implemented using this model must be able to infer user goals and build its own plans to provide the service requested by the user. For example, Cavazza et al. (2008) proposed an agent that generates an 'ideal' plan for the daily activities to be carried out by a human.

Following the same approach, Allen et al. (2007) presented PLOW, an intelligent conversational agent to assist the user in managing his daily tasks, whereas Eliasson (2007) implemented dialogue understanding and action planning in a

conversational agent set up in a robot, which was able to plan actions to obey the user.

The model based on agents takes into account that the dialogue manager carries out some reasoning to determine future actions (Turunen, 2004). This model relies on the collaboration of a number of intelligent agents to solve a specific problem or task. It is appropriate for complex tasks, for example, negotiation or troubleshooting, and typically employs mixed initiative for the dialogue management (McTear, 2004).

3.4.4 Statistical Approaches

Statistical (or data-based) approaches allow designing automatically the dialogue management strategy by learning a dialog model from a labelled dialogue corpus. The design is much more complicated in the case of the methods discussed above, which requires hand-crafting rules or plans. However, as these models can be trained on corpora of real human-computer dialogue, they explicitly model the variance in user behaviour that hand-written rules cannot cover.

The objective is to build systems which offer more robust performance, improved portability, better scalability and greater scope for adaptation (Schatzmann et al., 2006). Another advantage is in terms of the scalability, as the complexity of the dialogue manager can increase without causing problems for the agent's designer (Schatzmann et al., 2006). The drawback is that they require a considerable amount of data in order to properly compute the probabilities that decide the behaviour of the agent.

A number of techniques have been proposed in the literature following this approach. For example, Levin & Pieraccini (1997) defined a technique for learning dialogue strategies, which can be considered the antecedent of many posterior studies on reinforcement learning (Paek & Horvitz, 2004; Lemon et al., 2006b).

Williams & Young (2007) considered a spoken conversational agent as partially observable Markov decision process (POMDP). This process serves as a basis for optimisation the dialogue management and can integrate the uncertainty of the state of the dialogue in the form of statistical distributions.

Griol et al. (2008) presented a technique to develop a dialogue manager and learn optimal dialogue strategies from a labelled corpus acquired for the specific task. The answers of the conversational agent are generated using a classification process which considers the complete dialogue history. This technique was applied to develop the dialogue manager of an agent that provides railway information using spontaneous speech in Spanish (Griol et al., 2006).

3.4.5 VoiceXML

VoiceXML[1] is a standard language to access web applications by means of speech (McGlashan et al., 2004). The language is the result of the joint efforts of several companies and institutions (AT&T, IBM, Lucent, Motorola, etc) which make up the so-called VoiceXML Forum. The language has been designed to ease the creation of conversational agents employing audio, ASR, speech synthesis and recording, and mixed-initiative dialogues. The Florence dialogue manager (Fabbrizio & Lewis, 2004), developed by AT&T Labs, supports mixed initiative as well as different strategies for data confirmation and error correction.

There are two main models for dialogue management in VoiceXML. In the Augmented Transition Networks (ATN), the dialogue flow is represented by a set of states, transitions, conditions and variables. A transition to a specific state is selected when the conditions and prefixed actions have been carried out. The second strategy, called clarification flow controllers, defines the dialogue strategy using a hierarchical tree. The tree includes conditions that describe categories, topics and messages (prompts) to inform the user.

3.5 Embodied Conversational Agents (ECAs)

Many studies can be found in the literature regarding the analysis of facial expressions (Tian et al., 2003) and head movements (Morency et al., 2005; Maatman et al., 2005). The knowledge obtained from these analyses has been used as well to represent facial expressions and head movements of the so-called Embodied Conversational Agents (ECAs), which are typically implemented using FACS, MPEG-4 or XML-based languages.

FACS (Facial Action Coding System) is a comprehensive, notational system created by Ekman & Friesen (1978) with the goal to objectively describe facial activity. The system is based on several studies about the activity of facial muscles. It represents facial expressions by means of AUs (*action units)* which model the contraction of muscles (or of a set of them if they are somehow connected). Some AUs can operate in either side of the face, independently of the other side, in which case the user must specify "Left", "Right" or "Both". The combination of AUs can generate more than 7,000 facial expressions (Pelachaud et al., 2004). The direct manipulation of AUs can result difficult for non-experienced users, this is why a number of FACS-based animation toolkits have been developed (Patel & Willis, 1991).

MPEG-4 is a standard for compression of multimedia information that is being used on a variety of electronics products (Malatesta et al., 2009; Tekalp & Ostermann, 2000; Pandzic, 2002). The face models defined in MPEG-4 try to reproduce as faithfully as possible the visual manifestation of speech, the transmission of emotional information by means of the facial expressions, and the face of the speaker.

The standard defines 84 features points (FPs) located in a face model that describes a standard face. These points are used to define FAPs (Facial Animation Parameters) which calibrate facial models when different face players are used. MPEG-4 defines six high levels of expressions with two expression parameters (*viseme* and *expression*): joy, sadness, anger, fear, disgust and surprise. Better results are obtained using other low-level parameters. Each FAP corresponds to a FP and defines low-level deformations applicable to the FP with which it is associated. The FAPs represent a set of standard inputs that the animator can use. However, low-level parameters are not easy to use, and it is preferable to use tools that generate them from scripts written in a high-level language.

Several XML-based languages can be found in the literature to control the behaviour of ECAs. One important characteristic of these kinds of languages is the use of high-level primitives. For example, De Carolis et al. (2002) used AMPL (Affective Plan Markup Language) and DPML (Discourse Plan Markup Language) to control de behaviour of an ECA that has two components: *mind* and *body*. The mind represents the personality and intelligence of the agent, which generates the emotional response to the events occurring in its environment. The body represents the physical appearance. The interaction with the user is carried out using synchronized speech and facial expressions.

Tsutsui et al. (2000) used MPML (Multimodal Presentation Markup Language) to carry out multimodal presentations using ECAs, which can carry out a number of factions such as greet, point and explain. In addition to text and figures, the presentations can contain multimedia elements such as voice. One of the most important advantages of this language is that it is independent of the platform and browser employed by the user. Moreover, the multimedia elements can be played back in a number of tools or players.

Kopp et al. (2006) proposed a language called Behaviour Markup Language (BML), with elements and attributes to describe the behaviour of the conversational agent. For example, the element <head type="nod"/> is used to produce a nod. The elements that can be used are: head, torso, face, gaze, lips, body, gesture, legs and speech.

4. ARCHITECTURES

It is important to properly select the architecture to be used for implementing a conversational agent, since it should allow further enhancement of the agent or porting it from one application domain to another. We can find in the literature a number of architectures to implement conversational agents. In this section we discuss some of the most widespread (Galaxy Communicator, Open Agent Architecture, Blackboard and R-Flow) and comment on some other proposals.

4.1 Galaxy Communicator

Galaxy Communicator is a distributed, message-based, hub-centred architecture (Seneff et al., 1998). The main components are interconnected by means of a client-server architecture. This architecture that has been used to set up, among others, the MIT's Voyager and Jupiter agents (Glass et al., 1995; Zue et al., 2000).

4.2 Open Agent Architecture

The Open Agent Architecture (OAA) architecture was designed to ease the implementation of agent-based applications, enabling intelligent, cooperative, distributed, and multimodal agent-based user interfaces (Moran et al., 1997). The agents can be developed in several high-level languages (e.g. C or Java) and platforms (e.g. Windows and Solaris). The communication with other agents is possible using the Interagent Communication Language (ICL). The cooperation and communication between the agents is carried out by means of an agent called Facilitator. Several authors have used this architecture to implement conversational agents for a variety of application domains, including map-based tourist information (Moran et al., 1997), interaction with robots (Bos et al., 2003), and control of user movements in a 2D game (Corradini & Samuelsson, 2008).

4.3 Blackboard

The blackboard architecture was released considering principles of Artificial Intelligence. Its name denotes the metaphor of a group of expert people who work together and collaboratively around a blackboard to solve a complex problem. All the resources available are shared by the agents. Each agent can collaborate, generate new resources and use resources from other agents. A Facilitator agent controls the resources and acts as intermediary among the agents which compete to write in the blackboard, taking into account the relevance of the contribution of each agent.

This architecture has been used to implement a number of conversational agents. For example, Wasinger et al. (2003) used it to represent, analyse and make the fusion of multimodal information in a mobile pedestrian indoor/outdoor navigation system set up in a PDA device. Raux & Eskenazi (2007) implemented a new version of the Olympus framework for the development of conversational agents (Bohus et al., 2007). Within this new framework the information provided by a number of agents is combined and stored in the Interaction State, which is implemented by means of a blackboard.

Huang et al. (2007) also used the blackboard architecture to create the GECA platform, which uses XML messages for the interconnection of the components of a conversational agent. The platform uses a server that handles the management of a number of services, including service naming and message subscription and forwarding.

A variant of the blackboard architecture is the multi-blackboard architecture (Alexandersson & Becker, 2001). It was used, for instance, in the SmartKom conversational agent (Pfleger et al., 2002; Wahlster, 2006) to combine speech with not verbal modalities in order to help processing intelligible multimodal utterances (Kopp & Wachsmuth, 2004). More recently, Huang et al. (2008) have used this architecture to integrate components of an ECA. These components share data in the

blackboards by means of a subscribe-publish message passing mechanism. Each blackboard has its own manager, and the architecture includes a server responsible of the message subscription and naming services of the ECA.

4.4 R-Flow

R-Flow is an extensible XML-based architecture for multimodal conversational agents (Li et al., 2007). It is based on a recursive application of the Model-View-Controller (MVC) design. The structure is based on three layers: modality independent dialogue control, synchronization of logical modalities and physical presentation. Each one has been codified in different XML-based languages. State-Chart XML (SCXML) is used for dialogue control, SMIL (Synchronized Multimedia Integration Language) and EMMA (Extensible Multimodal Interface Language) based XM-Flow (Li et al., 2006) for modality synchronization and interpretation, and the physical presentation in a generic XML. The prototype presented in (Li et al., 2007) has been developed to manipulate Google map in a multimodal way.

4.5 Other Architectures

In addition to the architectures discussed above, which are amongst the most employed, it is possible to find other architectures in the literature. For example, Leßmann & Wachsmuth (2003) used the classical architecture Perceive-Reason-Act for the design of a conversational agent. The *Perceive* module handles the input information, which is collected by sensors (auditory, tactile and visual). The *Act* module generates the output information. Actions can be carried out by means of either *deliberative* or *reactive* behaviour. The component for deliberative behaviour is located in the *Reason* section of the figure. It uses knowledge about the domain updated by perceptions, and generates intentions employing a plan library, which represents what the agent wants to do next. The second way of generating an action is by means of the reactive behaviour, which is reserved for actions that do not need deliberation, for example, making the agent appear more lifelike.

Following a different approach, Wei and Rudnicky (2000) proposed an architecture based on a task decomposition and an expectation agenda. The agenda is a list of topics represented by handlers. A handler encapsulates the knowledge necessary for interacting with the user about a specific information slot. The agenda defines a "plan" for carrying out a specific task, which is represented as a specific order of handlers.

5. TOOLS FOR DEVELOPMENT

In this section we focus on tools for developing components of multimodal conversational agents, paying special attention to tools for automatic speech recognition, spoken language understanding, dialogue management, natural language generation, speech synthesis and embodied conversational agents.

5.1 Tools for Automatic Speech Recognition

The Hidden Markov Model Toolkit (HTK) was developed by Cambridge University (Young et al., 2000). It is free software for building and using Hidden Markov Models (HMMs). In the community of conversational agents this software is primarily used for ASR, but has been used for a number of applications including character recognition and DNA sequencing. It consists of a set of library modules and tools available in C source form that provide facilities for speech analysis, HMM training, testing and results analysis.

CMU Sphinx (Lee et al., 1990) describes a group of speech recognition systems developed at the Carnegie Mellon University. These include a series of speech recognizers (Sphinx 2 - 4) and an acoustic model trainer (SphinxTrain). Sphinx is a

continuous-speech, speaker-independent recognition system making use of HMMs and an n-gram statistical language model. Sphinx 2 focuses on real-time recognition suitable for speech-based applications and uses a semi-continuous representation for acoustic modeling. Sphinx 3 adopted the prevalent continuous HMM representation and has been used primarily for high-accuracy, non-real-time recognition. Sphinx 4 is written entirely in Java with the goal of providing a more flexible framework for research. PocketSphinx has been designed to run in real time on handhelds and be integrated with live applications.

Julius is a two-pass large vocabulary continuous speech recognition software for speech-based applications in Japanese (Lee & Kawahara, 2009). It is based on word 3-gram and context-dependent HMMs, and includes functionalities such as real-time accurate recognition, N-best and word graph outputs, confidence scoring, etc.

Sonic is a large vocabulary continuous speech recognition system developed by the University of Colorado. It is based on continuous density Hidden Markov acoustic models (Pellom & Hacioglu, 2003).

There is a number of proprietary software for ASR, including AT&T WATSON, Windows speech recognition system, IBM ViaVoice, Microsoft Speech API, Nuance Dragon NaturallySpeaking, MacSpeech, Loquendo ASR and Verbio ASR.

5.2 Tools for Spoken Language Understanding

The Carnegie Mellon Statistical Language Modeling Toolkit (CMU SLM) is a set of Unix tools designed to facilitate language modeling (Rosenfeld, 1995). The toolkit allows processing corpora of data (text strings) in order to obtain word frequency lists and vocabularies, word bigram and trigram counts, bigram and trigram-related statistics and a number of back-off bigram and trigram language models. Using these tools it is also possible to compute statistics such as perplexity, out-of-vocabulary words (OOV) and distribution of back-off cases.

The Natural Language Toolkit (NLTK) (Bird et al., 2008) is a suite of libraries and programs for symbolic and statistical natural language processing for the Python programming language.

Other tools include Phoenix, designed by the Carnegie Mellon University in combination with the Helios confidence annotation module (Ward & Issar, 1994), and Tina, developed by the MIT based on context free grammars, augmented transition networks, and lexical functional grammars (Seneff, 1989).

5.3 Tools for Dialogue Management

A number of toolkits for dialogue management can be found in the literature, which can be classified taking into account the model employed to represent the dialogue management, as was discussed in section 3.4.

5.3.1 Dialogue Management Based on Finite States

The Center for Spoken Language Understanding (CSLU) at the Oregon Health and Science University developed a graphical tool called CSLU Toolkit for the design of dialogue managers based on finite states (McTear, 1998).

Another tool for building agents based on finite state systems is the AT&T FSM library. It is a set of Unix tools for building, combining and optimizing weighted finite-state systems (Mohri, 1997). Some conversational agents based on finite states have been created under the SUNDIAL (Müller & Runge, 1993) and SUNSTAR projects (Nielsen & Baekgaard, 1992).

5.3.2 Dialogue Management Based on VoiceXML

There are currently many implementations developed in VoiceXML. For example, OpenVXI

is a portable open source VoiceXML interpreter available from Carnegie Mellon University and developed by SpeechWorks. It can be used free of charge in commercial applications and also allows the addition of proprietary modifications.

JVoiceXML is an open source VoiceXML interpreter for JAVA. Its main goal is to provide platform-independent implementation of conversational agents that can be used for free.

The OptimSys VoiceXML platform also allows the easy integration with ASR and TTS engines and telephony hardware of your choice.

BeVocal Café is a web-based VoiceXML development environment providing a VoiceXML interpreter that includes speaker verification, voice enrolment, XML data, pre-tuned grammars and professional audio.

Loquendo has developed a VoiceXML Interpreter integrated within the VoxNauta Platform. In addition, Loquendo Café provides developers with resources and tools to learn about creating speech-based applications.

Other tools include the following: Eloquant, HeyAnita, HP OpenCall Media platform, Intervoice's Omvia Media Server, Lucent MiLife VoiceXML Gateway, Motorola VoxGateway, Nuance VoiceXML platform, Vocalocity's platform, and Voxeo VoiceCenter IVR.

5.4 Tools for Natural Language Generation

Natural language generation is the process of obtaining texts in natural language from a non-linguistic representation. It is usually carried out in 5 steps: content organization, content distribution in sentences, lexicalization, generation of referential expressions, and linguistic realization. The simplest approach consists in using predefined text messages (e.g. error messages and warnings). Although intuitive, this approach completely lacks from any flexibility.

The next level of sophistication is template-based generation, in which the same message structure is produced with slight alterations. The template approach is used mainly for multi-sentence generation, particularly in applications which texts are fairly regular in structure such as some business reports. Rosetta (Oh & Rudnicky, 2000) is a toolkit developed by the CMU for language generation based on the latter approach.

5.5 Tools for Speech Synthesis

Text-to-speech (TTS) synthesizers transform a text string into an acoustic signal. A TTS system is composed of two components: front-end and back-end. The front-end transforms raw text containing symbols such as numbers and abbreviations into their equivalent words. It assigns phonetic transcriptions to each word, divides and marks the text into prosodic units, i.e. phrases, clauses and sentences. The back-end (often referred to as synthesizer) converts the symbolic linguistic representation obtained by the previous component into speech.

Festival (Clark et al., 2004) is a C++ general multi-lingual speech synthesis system developed at Centre for Speech Technology Research (CSTR) at the University of Edinburgh. It is distributed under a free software license and offers a number of APIs as well as an environment for development and research on speech synthesis. Supported languages include English, Spanish, Czech, Finnish, Italian, Polish and Russian.

FreeTTS (Walker et al., 2002) is an open source speech synthesis system written entirely in Java. It allows employing markers to specify when speech generation should not be interrupted, to concatenate speech, and to generate speech using different voices. FreeTTS is based upon CMU Flite (Festival-lite).

Some commercial systems for TTS are Cepstral, Loquendo TTS and Kalliope.

5.6 Tools for Embodied Conversational Agents

Xface (Balci, 2005) is an open source toolkit for generating and animating 3D talking heads. The toolkit relies on MPEG-4 Facial Animation Parameters (FAPs) and keyframe-based rendering driven by SMIL-Agent scripting language. All the components in the toolkit are independent of the operating system, and can be compiled with any ANSI C++ standard compliant compiler.

The CSLR's Conversational Agent Toolkit (CAT) (Cole et al., 2003) provides a set of modules and tools for research and development of advanced ECAs. These modules include an audio server, the Sonic speech recognition system, and the Phoenix natural language parser. The CU Animate toolkit (designed for research, development, control and real time rendering of 3D animated characters) is used for the design of the facial animation system.

Microsoft Agent toolkit (Walsh & Meade, 2003) includes animated characters, TTS engines, and speech recognition software. It is preinstalled in several versions of MS Windows and is as an ActiveX control that can be used by web pages. The speech engine is used by means of the Microsoft Speech API (SAPI). New Agent characters can be created using Microsoft's development tools, including the Agent Character Editor. Agents can be embedded in applications with Visual Basic and in web pages with VBScript.

Maxine (Seron et al., 2006) is an open source engine for embodied conversational agents developed by the University of Zaragoza (Spain). It enables interaction with the user by means of different channels, for example, text, voice, mouse and keyboard. The agent can gather information from the user and the environment (noise level in the room, position of the user to establish visual contact, image-based estimate of the user's emotional state, etc.). The agent can interact with the user by means of speech (in Spanish) and has its own emotional state, which depends on the relationship with the user.

Currently there are also several initiatives for the design of conversational agents and chatbots which are able to interact with the user in social networks and virtual worlds (Ieronutti & Chittaro, 2007; Hubal et al., 2008).

6. EVALUATION METHODOLOGIES

As conversational agents become more and more complex, it is necessary to develop new evaluation measures and methodologies to test their performance. The definition of new measures and procedures uniquely accepted by the scientific community for the assessment of agents presents many difficulties. In fact, this field can be considered to be still at an early development stage. In this section we firstly address general frameworks for evaluation and discuss evaluation types. Then, we describe briefly well-known approaches for the evaluation of conversational agents: PARADISE, PROMISE, CAS, WOz and simulation of user-agent interactions. Other approaches to the evaluation of multimodal conversational agents can be found in (Cassell et al., 2000; Bernsen, 2002).

6.1 General Frameworks for Evaluation

In recent years, various initiatives have been developed to define general frameworks that include the design and evaluation of conversational agents. In the United States one of the main projects was DARPA Communicator (Walker et al., 2001). Some examples in Europe are EAGLES (Expert Advisory Group on Language Engineering Standards) (King et al., 1996) ELSE (Paroubek & Blasband, 1999) and DISC (Bernsen et al., 1998).

Other European institutions that have focused on the study and definition of evaluation techniques are the following:

- COSCODA (Coordinating Committee on Speech Databases and Speech I/O Systems) is concerned with aspects related to the creation of multilingual databases.
- ELRA (European Language Resources Association) is focused on the collection and distribution of linguistic resources.
- SQUALE (Speech Recognition Quality Assessment for Linguistic Engineering) (Young et al., 1997) focused on the adaptation of the ARPA Large Vocabulary Continuous Speech Recognition paradigm (LVCSR) to multilingual contexts.

Two fundamental trends for the evaluation of conversational agents can be considered with regard these initiatives. On the one hand, the definition of quantitative measures to evaluate the quality of the agents (e.g., EAGLES and DARPA Communicator projects). On the other hand, proposals for the definition of qualitative and quantitative measures (e.g. ELSE and DISC projects).

The EAGLES evaluation group proposed a number of quantitative measures, which include: completion task, transaction success, system's response time and conciseness of agent's responses. It also proposed several qualitative measures, such as user satisfaction, agent's adaptation to new users and multimodality features. The group did not only provide insights on what aspects to evaluate, but also on how to carry out the evaluation and report results, setting up a set of parameters and methodology for homogeneous comparison between agents.

Similarly, the DISC project proposed aspects to be evaluated and criteria for evaluation. The methodology was based on templates and considers aspects regarding the life cycle of software.

LINTEST is a tool for the evaluation of conversational agents using dialogue corpora (Degerstedt & Jönsson, 2006). It allows two operation modes: batch and interactive. Using the former the evaluation generates a log file that includes the evaluation results. The latter allows a more detailed evaluation carried out during the interaction.

6.2 Evaluation Types and Measures

We can find in the literature many proposals to evaluate conversational agents. For example, Dybkjaer & Bernsen (2000) proposed a set of 15 criteria to ensure the usability of the agents: (1) use of the different modalities, (2) recognition of the user inputs, (3) coverage of user utterances regarding vocabulary and grammars, (4) voice quality of the agent, (5) generation of appropriate responses, (6) agent's feedback, (7) use of different dialogue initiatives for different dialogue tasks, (8) naturalness of the dialogue structure for different tasks, (9) domain coverage, (10) reasoning abilities of the agent, (11) guidance and help for the user during the interaction (12) features on error handling, (13) adaptation to differences between users, (14) existence of communication problems during the interaction, and (15) user satisfaction.

The evaluation measures can be either *objective* or *subjective*. The former are directly obtained from the interaction with the system, not including any kind of assessment made by developers or users. The latter includes an evaluation process typically carried out by the end users of the agent. For example, these measures were employed in the European project Trindi (Larsson et al., 1999).

The evaluation measures can also be classified taking into account how the computing of the evaluation scores is carried out (automatic or manual), or considering the influence on the overall quality of the system (positive or negative measures).

Taking into account the objective of the evaluation, two kinds of evaluation can also be distinguished: *black box* and *crystal box*. The former considers the overall performance of the agent, considering only its inputs and outputs. The latter focuses on the performance of agent's components separately, taking into account inputs and outputs

Table 1. Measures defined for the evaluation of the different modules of a conversational agent

Automatic Speech Recognition
Word Accuracy, Word Error Rate, Word Insertions Rate, Word Insertions Rate, Word Substitutions Rate, Sentence Accuracy
Natural Language Understanding
Percentage of words correctly understood, not covered or partially covered; Percentage of sentences correctly analyzed; Percentage of words outside the dictionary; Percentage of sentences whose final semantic representation is the same as the reference; Percentage of correct frame units, considering the actual frame units; Frame-level accuracy; Frame-level coverage
Dialogue Management
Strategies to recover from errors, to correct/direct user interaction, context management when there are multiple questions and answers associated with a scenario) (% correct responses,% of incorrect answers,% of half-answers, % of times the system works trying to solve a problem,% of times the user acts trying to solve a problem, etc.)
Natural Language Generation
Number of times the user requests a repetition of the reply provided by the system; User response time; Number of times the user does not answer; Rate of out of vocabulary words
Speech Synthesis
Intelligibility of synthetic speech and naturalness of the voice

of these modules. Table 1 summarizes the most commonly employed measures for the evaluation of the different modules of a conversational agent (San Segundo, 2004).

According to the reference taken for the evaluation of the conversational agent, we can distinguish several types of evaluation. In the *comparative* evaluation, different agents are developed in parallel with the same specifications by different research centers. This evaluation type has been usually used in projects funded by DARPA, e.g., DARPA Communicator. In the *temporary* evaluation the reference is the developed agent, and the goal is to make performance comparisons in several development stages. The *substitutive* evaluation compares the agent with another agent with the same capabilities previously developed, usually employing different technologies. The *initial* evaluation is employed when the reference agent is not available. It makes an estimation of performance *a priori* during the specification phase, and in subsequent evaluations considers the deviation from the expected performance.

6.3 PARADISE

PARADISE (PARAdigm for DIalogue System Evaluation) is one of the most employed proposals for globally evaluating the performance of conversational agents (Walker et al., 1998; Dybkjaer et al., 2004). It combines different features in a single function that measures the performance of the agent in direct correlation with user satisfaction. The main assumptions of the approach are two. Firstly, the main goal is to maximise user satisfaction. Secondly, task success and several dialogue costs (objective measures) can be used to predict user satisfaction. These two assumptions are interrelated as shown in the equation in Box 1.

The maximisation of user satisfaction is carried out by minimising dialogue costs and maximising task success. Dialogue costs are quantified by means of efficiency and quality measures. The most commonly used measures on task success are two. The first one is the Kappa factor, which is computed from a confusion matrix of the values of attributes exchanged between the user and the agent. The second measure is completion task, which is computed considering the number of times that the system correctly satisfies the user requests.

6.4 PROMISE

PROMISE (PROcedure for Multimodal Interactive System Evaluation) (Beringer et al., 2002) is an extension to multimodality of the PARADISE framework. This paradigm uses methods traditionally employed to evaluate spoken conversational agents, and specific methods to assess the characteristic properties of multimodal conversational agents, as for example, the combination of gestures

Box 1.

$$User\,Satisfaction = \propto N\left(Task\,Success\right) - \sum_{i=1}^{N} \omega_i N(Costs\,of\,the\,dialogue)$$

and speech in the input, the combination of speech and graphics in the output, etc.

According to this procedure, the evaluation is carried out by defining a number of qualitative and quantitative measures (called costs) that have an associated weight. Instead of using a linear regression (as in the case of the PARADISE), PROMISE employs a calculated peer Pearson correlation "user - satisfaction cost", some of these objectives costs and other subjective. For the evaluation, test users interact with the system and fill in a questionnaire which includes subjective costs. Some of these costs are equivalent to those used in the procedure PARADISE, while others are used to treat specifically multimodality and behaviour of non-cooperative users.

The most important efficiency and quality measures defined for these models are the average time needed to complete a task, average time per turn, average number of turns per task, minimum time to complete a specific task, types of confirmations used by the system, number of words correctly recognized per turn, rate of correct semantic concepts, percentage of correctly corrected errors, time employed for the user and the system to answer, number of times that the user does not answer, number of times that the user requires a repetition or ask for help, number of times that the user interrupts the system prompt, etc.

6.5 CAS

The CAS (Common Answer Specification) approach evaluates the performance of the conversational agent by comparing the responses of the agent with canonical responses extracted from a database (Boisen et al., 1989). This allows automatic evaluation of the agent once the principles for generating the reference responses have been defined, and a labelled corpus for the specific task is available. In addition, it allows the direct comparison of agents.

However, the evaluation with this approach is very limited since it is carried out at the sentence level only, i.e. comparing each agent's response with the canonical response. Moreover, it is not possible to distinguish between partially correct responses and totally wrong ones. Therefore, it does not allow detecting or correcting errors, or evaluating the quality of the responses. Among others, this approach has been used in the ARPA projects to evaluate agents designed for the ATIS domain (Air Travel Information Systems).

6.6 Wizard of Oz

The Wizard of Oz technique (WOz) is usually employed to emulate the system performance, as was discussed in section 3.1 (Fraser & Gilbert, 1991). To do this, the approach typically employs a set of scenarios that define the goals the user must try to achieve during the interaction with the conversational agent. The interaction is stored in log files, containing additional information such as user utterances, speech recognition results, semantic representations obtained, agent responses, and time required by the user to answer each agent's prompt (Webb et al., 2010).

A questionnaire is used to consider the user acceptation of the different functionalities of the agent, for example, quality of synthesised speech, ease for error correction, interactivity and friendliness. Taking into account the dialogue logs and questionnaires, it is possible to compute a set of

measures that allow to quantitative evaluation of the agent. Some of these measures include: time required to accomplish the scenario goals, number of user questions correctly answered by the agent and user satisfaction.

6.7 User Simulation

A technique that has attracted increasing interest in the last decade for the evaluation of conversational agents is based on the automatic generation of dialogues between the agents and an additional module, called user simulator, which represents user interactions (Zukerman & Litman, 2001; López-Cózar et al, 2003; Schatzmann et al., 2006; Griol et al., 2009). The simulator makes it possible to generate a large number of dialogues in a very simple way. Therefore, this technique reduces the time and effort that would be needed for evaluating an agent each time it is modified in order to improve performance.

The construction of user models based on statistical methods has provided interesting and well-founded results in recent years and is currently a growing research area. In terms of user simulation, the goal is to obtain a probabilistic user model from the analysis of a corpus of human-computer interaction, which can be employed for setting up the user simulator (Pietquin & Dutoit, 2005; Cuayáhuitl et al., 2005; Schatzmann & Young, 2009).

7. A CASE OF STUDY: HADA-DS

Our work within the HADA project (Adaptive Hypermedia for Attention to Different User Types in Ambient Intelligence Environments) is concerned with setting up a multimodal conversational agent, termed HADA-DS, to assist professors and students in some of their daily activities within a University's Faculty (López-Cózar et al., 2011). The agent works in three different places of the Faculty: Library, Professors' Offices and Classrooms. Our goal is that by using the agent, professors may interact more easily with devices in their environment, e.g. classroom beamers or lights. Moreover, students may receive different types of information depending on their localisation within the environment.

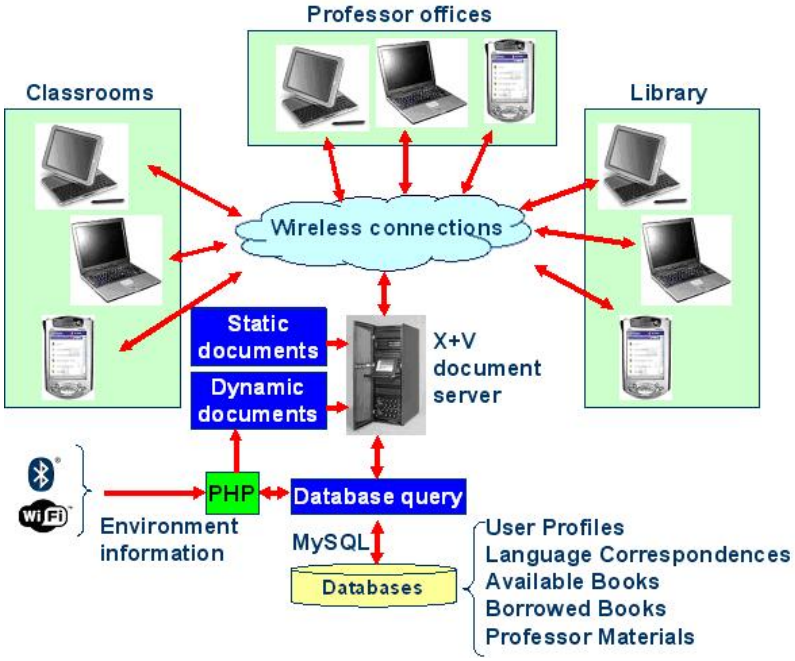
The agent allows multimodal interaction for the user input, namely, using speech, keyboard or mouse. For example, a student can ask for information about available books on a particular subject by either speaking the subject, selecting it on the screen of his/her mobile device, or writing the subject in a form field. Since the agent's output is multimodal as well, a spoken message for this request may indicate that the requested information is available on the screen. The agent does allow combining data provided by different information sources in just one interaction, but allows combining data provided by the user in different dialogue turns.

Figure 1 shows the architecture of the agent, which is comprised of an XHTML+Voice (X+V) document server connected with the users' mobile devices (tablet PCs, laptop computers and PDAs) by means of wireless connections.

7.1 XHTML+Voice Documents

The logic of the agent is implemented by means of a set of X+V documents. Some of these documents are stored in the document server, while others are dynamically created using PHP programs that take into account features stored in the user profile (e.g. user gender and preferred interaction language), as well as data extracted from databases. X+V documents are comprised of forms, the fields of which are filled in with the user input provided via speech, text or mouse clicks. To visualise the documents, users must run in their communication device the Opera browser[2], which enables multimodal interaction using voice, text, mouse clicks and graphics.

Figure 1. Architecture of the HADA-DS conversational agent



7.1.1 Speech-Based Interaction

Automatic speech recognition is carried out by the Opera browser's built-in recogniser. In our setting the recognition is based on a tap-&-talk method, i.e. the user must click and hold a microphone icon or press a key while s/he speaks to the agent. Speech recognition and understanding is carried out using JSGF (Java Speech Grammar Format) grammars that are used either at form or field level. Some of these grammars are static, while others are dynamically created by means of PHP programs that query databases and include the obtained data in the grammars (e.g. book titles). For example, using the grammar to recognise book queries, if a user utters the sentence *I need books about Maths please*, the agent fills in the form field *subject* with the word *Maths*.

The recognition grammars used to handle book queries must be updated as the library catalogue changes, so that they are compiled dynamically using the contents of the *Available Books* database. To update these grammars we have implemented a PHP program that carries out two tasks. Firstly, it queries databases using MySQL functions and obtains data from available books, such as titles, authors or subjects. Secondly, it creates the grammars to recognise complete sentences as well as isolated data items (e.g. title, authors or subjects) using the information gathered in the first step.

In the system output, speech synthesis is carried out by means of sentence patterns included into the <prompt> … </prompt> labels typically used in VoiceXML[3]. These sentences are transformed into voice by a Text-To-Speech (TTS) process using the Opera browser's built-in speech synthesiser. Some of these sentences are fixed, while others are created at run-time considering the user type (professor or student), the user gender (necessary to create sentences in Spanish appropriately) and data extracted from databases.

7.1.2 GUI-Based Interaction

In the system input, the visual interaction is used to obtain data from the user via form fields and selection buttons typically used in XHTML. In the system output, the visual interaction is used to provide data extracted from databases (e.g. list of available books) and information about the current user's name and type.

7.1.3 Connection of Both Interfaces

The connection between the speech- and GUI-based interfaces is carried out using event handlers, which are placed at the body section of the X+V documents. We use several types of event handlers available in X+V. For example, when the document used to enter book queries is loaded into the browser, the event onload is thrown and, in response, a VoiceXML form called initial_vform is executed to handle this event.

XHTML+Voice allows that a user utterance can fill in several form fields in one interaction (mixed-initiative interaction strategy). To do so, we use a <vxml:initial name="initial_vform"> …</initial> section, typically employed in VoiceXML, which allows recognising the user utterance using a form level grammar. Thus, for example, for the book query document the system generates the message *Please enter a book query* and the user can utter a variable number of data items (e.g. authors; authors and publication year; authors, publication year and subjects; etc.). We also use the ev:event="onclick" event, which is thrown when the user clicks on a form field. The handler for this event is VoiceXML code to obtain the value for that particular form field.

7.2 Agent's Interaction with the Environment

Our goal is that the agent-user interaction can be carried out in such a way that the location in which the user is interacting at every moment (e.g. in a professor's office) can be taken into account by the conversational agent without the user being concerned. By doing so we expect to enable a more intelligent behaviour of the agent. For example, if a professor says to the agent: *"Switch on the light"* when he is in a room where there are several lights, the agent should ask which light the user is referring to.

Obviously, the agent should not ask this question if the user is a room where there is one light only. To achieve this goal we are using RFID (Radio Frequency IDentification) technology. Each user has one RFID card that identifies him/her, and there are a number of RFID readers in different places of our intelligent environment (Faculty) for user localisation. At the time of writing, we are working in the setting up of a middleware layer, more specifically a *blackboard* (Alamán et al., 2001), to receive information from the RFID readers and the devices in the environment. Using this middleware the agent will operate the devices (e.g. switching them on/off) by changing their status in the blackboard.

8. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this chapter we have discussed benefits and disadvantages of multimodal interaction for conversational agents. We have addressed implementation techniques, discussing the Wizard of Oz and the System-in-the-loop methods. Moreover, we have discussed the *fusion* and *fission* of multimodal information, and focused on the implementation of the dialogue manager.

Later on, the chapter has addressed Embodied Conversational Agents (ECAs) and agent architectures, focusing on Galaxy Communicator, OAA, Blackboard and R-Flow. It has provided as well a description of tools to develop components of the agents, focusing mainly on automatic speech recognition, spoken language understanding, dia-

logue management, natural language generation, speech synthesis and ECAs.

We have discussed as well evaluation methodologies, and focused on general frameworks, evaluation types and measures, as well as a number of evaluation techniques (PARADISE, PROMISE, Common Answer Specification, Wizard of Oz and user simulation). Finally, the chapter has discussed our current work in the development of a multimodal conversational agent to assist professors and students in some of their daily activities within a University's Faculty.

The development of multimodal conversational agents is a very active research topic. The design and performance of these agents is very complex, not only because of the complexity of the different technologies involved, but also because of the required interconnection of very different technologies. Hence, additional work is needed in several directions to make these systems more usable by a wider range of potential users. For example, in terms of dialogue management, more studies are needed to set up more adaptive techniques, which learn user preferences and adapt the agent's behaviour accordingly.

The development of *emotional* conversational agents represents another line of research, which relies on the fact that emotions play a very important role in the rational decision-making, perception and human-to-human interaction. From a general point of view, emotionally-dependent dialogue management strategies must take into account that humans usually exchange their intentions using both verbal and non-verbal information. More information on the advances made in this line of research can be read in Chapter 9 of this book.

The development of *social* dialogue strategies is another research direction. It relies on the fact that in human-to-human interaction people do not only speak about topics concerned with the task at hand, but also about other topics and especially at the beginning of the conversation, for example, weather conditions, family or current news. This off-talk typically human dialogue, as can be read in Chapter 6, could also be used to improve the human-computer interaction. Hence, additional efforts must be made by the research community in order to make conversational agents more humanlike by designing dialogue strategies based on this kind of very genuine human behaviour.

ACKNOWLEDGMENT

This research has been funded by the Spanish project HADA TIN2007-64718.

REFERENCES

Alamán, X., Haya, P., & Montoro, G. (2001). *El proyecto InterAct: Una arquitectura de pizarra para la implementación de Entornos Activos* (pp. 72–73). Proc. of Interacción Persona-Ordenador.

Alexandersson, J., & Becker, T. (2001). Overlay as the basic operation for discourse processing in a multimodal dialogue system. *Proc. of IJCAI*.

Allen, J. (1995). *Natural language understanding*. The Benjamin/Cummings Publishing Company Inc.

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., & Stent, A. (2001). Towards conversational human-computer interaction. *AI Magazine*, *22*(4), 27–37.

Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., & Taysom, W. (2007). PLOW: A collaborative task learning agent. *Proc. of AAAI*, (pp. 22-26).

Balci, K. (2005). XfaceEd: Authoring tool for embodied conversational agents. *Proc. of ICMI*, (pp. 208-213).

Batliner, A., Fischer, K., Huber, R., Spliker, J., & Nöth, E. (2003). How to find trouble in communication. *Speech Communication*, *40*, 117–143. doi:10.1016/S0167-6393(02)00079-1

Beringer, N., Kartal, U., Louka, K., Schiel, F., & Türk, U. (2002). PROMISE - a procedure for multimodal interactive system evaluation. *Proc. of LREC Workshop on Multimodal Resources and Multimodal Systems Evaluatio*n, (pp. 77–80).

Bernsen, N. O. (2002). *Multimodality in language and speech systems - from theory to design support tool* (pp. 93–148). Kluwer Academic Publishers.

Bernsen, N. O., Dybkjaer, L., Carlson, R., Chase, L., Dahlback, N., Failenschmid, K., et al. Paroubek, P. (1998). The DISC approach to spoken language system development and evaluation. *Proc. of the First International Conference on Language Resources and Evaluation (LREC)*, (pp. 185-189).

Beveridge, M., & Fox, J. (2006). Automatic generation of spoken dialogue from medical plans and ontologies. *Biomedical Informatics*, *39*(5), 482–499. doi:10.1016/j.jbi.2005.12.008

Bird, S., Klein, E., Loper, E., & Baldridge, J. (2008). Multidisciplinary instruction with the Natural Language Toolkit. *Proc. of the Third ACL Workshop on Issues in Teaching Computational Linguistics*, (pp. 62-70).

Bohus, D., Raux, A., Harris, T., Eskenazi, M., & Rudnicky, A. (2007). Olympus: An open-source framework for conversational spoken language interface research. *Proc. of HLT-NAACL.*

Boisen, S., Ramshaw, L., Ayuso, D., & Bates, M. (1989). A proposal for SLS evaluation. *Proc. of the Workshop on Speech and Natural Language*, ACL Human Language Technology Conference, (pp. 135-146).

Bos, J., Klein, E., & Oka, T. (2003). Meaningful conversation with a mobile robot, *Proc. of EACL*, (pp. 71-74).

Carpenter, R. (1992). *The logic of typed features structures*. Cambridge University Press. doi:10.1017/CBO9780511530098

Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhálmsson, H., & Yan, H. (1999). *Embodiment in conversational interfaces: Rea* (pp. 520–527). Proc. of Computer-Human Interaction.

Cassell, J., Sullivan, J., Prevost, S., & Churchill, E. F. (Eds.). (2000). *Embodied conversational agents*. The MIT Press.

Catizone, R., Setzer, A., & Wilks, Y. (2003). Multimodal dialogue management in the COMIC project. *Proc. of EACL Workshop on Dialogue Systems: Interaction, Adaptation, and Styles of Management*, (pp. 25-34).

Cavazza, M., Smith, C., Charlton, D., Zhang, L., Turunen, M., & Hakulinen, J. (2008). A companion ECA with planning and activity modelling. *Proc. of AAMAS.*

Clark, R., Richmond, K., & King, S. (2004). Festival 2 - build your own general purpose unit selection speech synthesizer. *Proc. of 5th ISCA Workshop on Speech Synthesis*, (pp. 173–178).

Cole, R., Mariani, J., Uszkoreit, H., Varile, G. B., Zaenen, A., Zampolli, A., & Zue, V. (Eds.). (1997). *Survey of the state of the art in human language technology*. Cambridge University Press.

Cole, R., Van Vuuren, S., Pellom, B., Hacioglu, K., Ma, J., Movellan, J., et al. Wade-stein, D. (2003). Perceptive animated interfaces: First steps toward a new paradigm for human-computer interaction. *Proc. of the IEEE Special Issue on Multimodal Human Computer Interface*, (pp. 1391-1405).

Corradini, A., & Samuelsson, C. (2008). A generic spoken dialogue manager applied to an interactive 2D game. In E. André, L. Dybkjær, W. Minker, H. Neumann, R. Pieraccini, & M. Weber (Eds.) PIT 2008. *LNCS (LNAI)*, vol. 5078, (pp. 3–13).

Cuayáhuitl, H., Renals, S., Lemon, O., & Shimodaira, H. (2005). Human-computer dialogue simulation using Hidden Markov models. *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (pp. 290-295).

De Carolis, B., Carofiglio, V., & Bilvi, M. M., & Pelachaud, C. (2002). APML, a mark-up language for believable behavior generation. *Proc. of AAMAS*.

Degerstedt, L., & Jönsson, A. (2006). LinTest, a development tool for testing dialogue systems. *Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP),* (pp. 489-492).

Den, E., Boves, L., Lamel, L., & Baggia, P. (1999). *Overview of the ARISE project* (pp. 1527–1530). Proc. of Eurospeech.

Doherty, P., Granlund, G., Kuchcinski, K., Sandewall, E., Nordberg, K., Skarman, E., & Wiklund, J. (1998). The WITAS unmanned aerial vehicle project. *Proc. of the 14th European Conference on Artificial Intelligence (ECAI)*, (pp. 747-755).

Dybkjaer, L., & Bernsen, N. (2000). Usability issues in spoken language dialogue systems. *Natural Language Engineering*, *6*, 243–271. doi:10.1017/S1351324900002461

Dybkjaer, L., Bernsen, N., & Minker, W. (2004). Evaluation and usability of multimodal spoken language dialogue systems. *Speech Communication*, *43*, 33–54. doi:10.1016/j.specom.2004.02.001

Ekman, P., & Friesen, W. (1978). *Facial action coding system*. Consulting Psychologist Press.

Eliasson, K. (2007). Case-based techniques used for dialogue understanding and planning in a human-robot dialogue system. *Proc. of IJCAI*, (pp. 1600-1605).

Emele, M. C. (1994). The typed feature structure representation formalism. *Proc. of the International Workshop on Sharable Natural Language Resources*.

Fabbrizio, G., & Lewis, C. (2004). Florence: A dialogue manager framework for spoken dialogue systems. *Proc. of International Conference on Spoken Language Processing (ICSLP)*, (pp. 3065-3068).

Faure, C., & Julia, L. (1993). Interaction hommemachine par la parole et le geste pour l'édition de documents. *Proc. International Conference on Real and Virtual Worlds*, (pp. 171-180).

Forbes-Riley, K., & Litman, D. (2011). Designing and evaluating a wizarded uncertainty-adaptive spoken dialogue tutoring system. *Computer Speech & Language*, *25*(1), 105–126. doi:10.1016/j.csl.2009.12.002

Fraikin, F., & Leonhardt, T. (2002). *From requirements to analysis with capture and replay tools. PI-R 1/02*. Software Engineering Group, Department of Computer Science, Darmstadt University of Technology.

Fraser, N., & Gilbert, G. (1991). Simulating speech systems. *Computer Speech & Language*, *5*, 81–99. doi:10.1016/0885-2308(91)90019-M

Glass, J., Flammia, G., Goodine, D., Phillips, M., Polifroni, J., & Sakai, S. (1995). Multilingual spoken-language understanding in the MIT Voyager system. *Speech Communication*, *17*(1-2), 1–18. doi:10.1016/0167-6393(95)00008-C

Goddeau, D., Meng, H., Polifroni, J., Seneff, S., & Busayapongchai, S. (1996). A form-based dialogue manager for spoken language applications. *Proc. of International Conference on Spoken Language Processing (ICSLP)*, (pp. 701-704).

Graesser, A. C., VanLehn, K., Rose, C., Jordan, P., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, *22*, 39–51.

Griol, D., Callejas, Z., & López-Cózar, R. (2009). A comparison between dialogue corpora acquired with real and simulated users. *Proc. of the 10th Annual Meeting of the Special interest Group on Discourse and Dialogue (SIGDIAL 2009)*, (pp. 326-332).

Griol, D., Hurtado, L. F., Segarra, E., & Sanchis, E. (2008). A statistical approach to spoken dialogue systems design and evaluation. *Speech Communication*, *50*(8-9), 666–682. doi:10.1016/j.specom.2008.04.001

Griol, D., Torres, F., Hurtado, L., Grau, S., García, F., Sanchis, E., & Segarra, E. (2006). A dialogue system for the DIHANA project. *Proc. of SPECOM*, (pp. 131-136).

Huang, C., Xu, P., & Zhang, X. Zhao, S., Huang, T., & Xu, B. (1999). LODESTAR: A Mandarin spoken dialogue system for travel information retrieval. *Proc. of Eurospeech*, (pp. 1159-1162).

Huang, H., Cerekovic, A., Pandzic, I., Nakano, Y., & Nishida, T. (2007). A script driven multimodal embodied conversational agent based on a generic framework. *Proc. of IVA*, (pp. 381–382).

Huang, H.-H., Cerekovic, A., Nakano, Y., Pandzic, I. S., & Nishida, T. (2008). The design of a generic framework for integrating ECA components. *Proc. of AAMAS*, (pp. 128–135).

Hubal, R. C., Fishbein, D. H., Sheppard, M. S., Paschall, M. J., Eldreth, D. L., & Hyde, C. T. (2008). How do varied populations interact with embodied conversational agents? Findings from inner-city adolescents and prisoners. *Computers in Human Behavior*, *24*(3), 1104–1138. doi:10.1016/j.chb.2007.03.010

Ibrahim, A., & Johansson, P. (2002). Multimodal dialogue systems for interactive TV applications. *Proc. of 4th IEEE Int. Conf. on Multimodal Interfaces*, (pp. 117-122).

Ieronutti, L., & Chittaro, L. (2007). Employing virtual humans for education and training in X3D/VRML worlds. *Computers & Education*, *49*(1), 93–109. doi:10.1016/j.compedu.2005.06.007

Johnson, W. L., & Valente, A. (2008). Tactical language and culture training systems: Using Artificial Intelligence to teach foreign languages and cultures. *Proc. IAAI*, (pp. 1632-1639).

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., et al. Maloor, P. (2002). MATCH: An architecture for multimodal dialogue systems. *Proc. of 40th Annual Meeting of the ACL*, (pp. 376-383).

King, M., Maegaard, B., Schutz, J., & des Tombes, L. (1996). *EAGLES - Evaluation of Natural Language Processing Systems*, (Final report, EAG-EWG-PR.2).

Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., et al. (2006). Towards a common framework for multimodal generation in ECAs: The behavior markup language. *Proc. of 6th International Conference on Intelligent Virtual Agents*, (pp. 205-217).

Kopp, S., & Wachsmuth, I. (2004). Synthesizing multimodal utterances for conversational agents. *Computer Animation and Virtual Worlds*, *15*(1), 39–52. doi:10.1002/cav.6

Kuppevelt, J., & Dybkajer, L. (Eds.). (2005). *Advances in natural multimodal dialogue systems*. Springer. doi:10.1007/1-4020-3933-6

Larsson, S., Berman, A., Bos, J., Grönqvist, L., & Junglöf, P. (1999). *A model of dialogue moves and information state revision. Technical Report, D5.1 Trindi*. Task Oriented Instructional Dialogue.

Lee, A., & Kawahara, T. (2009). *Recent development of open-source speech recognition engine Julius. Proc of. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. APSIPA ASC.

Lee, K., Hon, H., & Reddy, R. (1990). *An overview of the SPHINX speech recognition system. Readings in Speech Recognition* (pp. 600–610). Morgan Kaufmann Publishers.

Lemon, O., Georgila, K., & Henderson, J. (2006b). Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: The TALK TownInfo evaluation. *Proc. of IEEE-ACL*, (pp. 178–181).

Lemon, O., Georgila, K., Henderson, J., & Stuttle, M. (2006a). An ISU dialogue system exhibiting reinforcement learning of dialogue policies: Generic slot-filling in the TALK in-car system. *Proc. of EACL*.

Leßmann, N., & Wachsmuth, I. (2003). A cognitively motivated architecture for an anthropomorphic artificial communicator. *Proc. of ICCM-5*, (pp. 277- 278).

Levin, E., & Pieraccini, R. (1997). *A stochastic model of computer-human interaction for learning dialogue strategies* (pp. 1883–1886). Proc. of Eurospeech.

Li, L., Cao, F., Chou, W., & Liu, F. (2006). XM-flow: An extensible micro-flow for multimodal interaction. *Proc. of MMSP*, (pp. 497-500).

Li, L., Li, L., Chou, W., & Liu, F. (2007). R-Flow: An extensible XML based multimodal dialogue system architecture. *Proc. of MMSP,* (pp. 86-89).

Liu, J., Wang, J., & Wang, C. (2006). Spoken language understanding in dialog systems for Olympic game information. *Proc. of IEEE Int. Conf. on Industrial Informatics*, (pp. 1042-1045).

López-Cózar, R., Ábalos, N., Espejo, G., Griol, D., Callejas, Z. (2011). Using ambient intelligence information in a multimodal dialogue system. *Journal of Ambient Intelligence and Smart Environments*. In printing.

López-Cózar, R., & Araki, M. (2005). *Spoken, multilingual and multimodal dialogue systems: Development and assessment*. Wiley.

López-Cózar, R., de la Torre, A., Segura, J., & Rubio, A. (2003). Assessment of dialogue systems by means of a new simulation technique. *Speech Communication*, *40*, 387–407. doi:10.1016/S0167-6393(02)00126-7

Maatman, R. M., Gratch, J., & Marsella, S. (2005). Natural behavior of a listening agent. *Proc. of IVA*, (pp. 25-36).

Mairesse, F., Gasic, M., Jurcicek, F., Keizer, S., Thomson, B., Yu, K., & Young, S. (2009). Spoken language understanding from unaligned data using discriminative classification models. *Proc. of ICASSP*, (pp. 4749-4752).

Malaka, R., Haeusseler, J., & Aras, H. (2004). SmartKom mobile: Intelligent ubiquitous user interaction. *Proc. of 9th Int. Conf. on Intelligent User Interfaces*, (pp. 310-312).

Malatesta, L., Raouzaiou, A. K., Karpouzis, K., & Kollias, S. D. (2009). Towards modeling embodied conversational agent character profiles using appraisal theory predictions in expression synthesis. *Applied Intelligence*, *30*(1), 58–64. doi:10.1007/s10489-007-0076-9

Maragoudakis, M. (2007). MeteoBayes: Effective plan recognition in a weather dialogue system. *IEEE Intelligent Systems*, *22*(1), 66–77. doi:10.1109/MIS.2007.14

Mayfield, L., & Burger, S. (1999). Eliciting natural speech from non-native users: Collecting speech data for LVCSR. *Proc. of ACL-IALL*.

McGlashan, S., Burnett, D. C., Carter, J., Danielsen, P., Ferrans, J., Hunt, A.,... Tryphonas, S. (2004). *Voice extensible markup language* (VoiceXML). W3C.

McTear, M. F. (1998). Modelling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit. *Proc. of ICSLP*, (pp. 1223–1226).

McTear, M. F. (2004). *Spoken dialogue technology. Toward the conversational user interface*. Springer.

Meza-Ruiz, I. V., Riedel, S., & Lemon, O. (2008). Accurate statistical spoken language understanding from limited development resources. *Proc. of ICASSP*.

Minker, W. (1998). Stochastic versus rule-based speech understanding for information retrieval. *Speech Communication*, *25*(4), 223–247. doi:10.1016/S0167-6393(98)00038-7

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, *23*(2), 269–311.

Moran, D. B., Cheyer, A. J., Julia, L. E., Martin, D. L., & Park, S. (1997). Multimodal user interface in the open agent architecture. *Proc. of ACM*, (pp. 61–68).

Morency, L. P., Sidner, C., Lee, C., & Darrell, T. (2005). Contextual recognition of head gestures. *Proc. of ICMI*, (pp. 18-24).

Müller, C., & Runge, F. (1993). *Dialogue design principles - key for usability of voice processing* (pp. 943–946). Proc. of Eurospeech.

Müller, J., Poller, P., & Tschernomas, V. (2003). A multimodal fission approach with a presentation agent in the dialog system SmartKom. *LNCS*, *2821*, 633–645.

Nielsen, P. B., & Baekgaard, A. (1992). Experience with a dialogue description formalism for realistic applications. *Proc. of International Conference on Spoken Language Processing (ICSLP)*, (pp. 719-722).

Nigay, L., & Coutaz, J. (1993). A design space for multimodal systems: Concurrent processing and data fusion. *Proc. of ACM CHI Conf. on Human Factors in Computing Systems*, (pp. 172-178).

Nigay, L., & Coutaz, J. (1995). A generic platform for addressing the multimodal challenge. *Proc. of ACM CHI*, (pp. 98-105).

Oh, A. H., & Rudnicky, A. (2000). Stochastic language generation for spoken dialogue systems. *Proc. of ANLP/NAACL workshop on conversational systems*, (pp. 27-32).

Paek, T., & Horvitz, E. (2004). Optimizing automated call routing by integrating spoken dialogue models with queuing models. *Proc. of HLT-NAACL*, 41-48.

Pandzic, I. S. (2002). Facial animation framework for the web and mobile platforms. *Proc. of Web3D Symposium*, (pp. 27-34).

Paroubek, P., & Blasband, M. (1999). *A blueprint for a general infrastructure for natural language processing systems evaluation using semi-automatic quantitative black box approach in a multilingual environment.* ELSE project Executive Summary.

Patel, M., & Willis, P. G. (1991). *FACES–The Facial Animation, Construction and Editing System* (pp. 33–45). Proc. of Eurographics.

Pelachaud, C., Maya, V., & Lamolle, M. (2004). Representation of expressivity for embodied conversational agents. *Proc. of Workshop Balanced Perception and Action, 3rd Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*.

Pellom, B., & Hacioglu, K. (2003). Recent improvements in the CU Sonic ASR System for noisy speech. *Proc. of ICASSP*.

Petrelli, D., De Angeli, A., Gerbino, W., & Cassano, G. (1997). Referring in multimodal systems: The importance of user expertise and system features. *Proc. ACL-EACL*, (pp. 14-19).

Pfleger, N., Alexandersson, J., & Becker, T. (2002). Scoring functions for overlay and their application in discourse processing. *Proc. of KONVENS*.

Pietquin, O., & Dutoit, T. (2005). A probabilistic framework for dialogue simulation and optimal strategy learning. *IEEE Transactions on Speech and Audio Processing, Special Issue on Data Mining of Speech*. *Audio and Dialog*, *14*, 589–599.

Potamianos, A., Ammicht, E., & Fosler-Lussier, E. (2003). Modality tracking in the Multimodal Bell Labs Communicator. *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (pp. 192-197).

Rabiner, L. R., & Juang, B. H. (1993). *Fundamentals of speech recognition*. Prentice-Hall.

Raux, A., & Eskenazi, M. (2007). A multi-layer architecture for semi-synchronous event-driven dialogue management. *Proc. of ASRU*.

Reithinger, N., & Sonntag, D. (2005). *An integration framework for a mobile multimodal dialogue system accessing the Semantic Web* (pp. 841–844). Proc. of Interspeech.

Rosenfeld, R. (1995). The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation. *Proc. of ARPA Spoken Language Systems Technology Workshop*.

Salber, D., & Coutaz, J. (1993). Applying the Wizard of Oz technique to the study of multimodal systems. *Proc. of EWHCI*, (pp. 219-230).

San Segundo, R. (2004). *La evaluación objetiva de sistemas de diálogo. Proc of. Curso de Tecnologías Lingüísticas*. Fundación Duques de Soria.

Schatzmann, J., Weilhammer, K., Stuttle, M., & Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, *21*(2), 97–126. doi:10.1017/S0269888906000944

Schatzmann, J., & Young, S. (2009). The hidden agenda user simulation model. *IEEE Trans. Audio. Speech and Language Processing*, *17*(4), 733–747. doi:10.1109/TASL.2008.2012071

Seneff, S. (1989). TINA: A probabilistic syntactic parser for speech understanding systems. *Proc. of ACL Workshop on Speech and Natural Language*, (pp. 168-178).

Seneff, S., & Hurley, E. Lau, R., Pao, C., Schmid, P., & Zue, V. (1998). Galaxy-II: A reference architecture for conversational system development. *Proc. of ICSLP*, (pp. 931-934).

Seron, F., Baldassarri, S., & Cerezo, E. (2006). MaxinePPT: Using 3D virtual characters for natural interaction. *Proc. of 2nd International Workshop on Ubiquitous Computing and Ambient Intelligence*, (pp. 241-250).

Steininger, S., Rabold, S., Dioubina, O., & Schiel, F. (2002). Development of the user-state conventions for the multimodal corpus in SmartKom. *Proc. of 3rd International Conference on Language Resources and Evaluation*.

Stent, A., Dowding, J., Gawron, J. M., Bratt, E., & Moore, R. (1999). The CommandTalk spoken dialogue system. *Proc. of 37th Annual Meeting of the ACL*, (pp. 183-190).

Takeuchi, A., & Nagao, K. (1995). Situated facial displays: Towards social interaction. *Proc. of SIGCHI*, (pp. 450-454).

Tekalp, M. A., & Ostermann, J. (2000). *Face and 2-D mesh animation in MPEG-4. Image Communication Journal, Tutorial Issue on MPEG-4 Standard*. Elsevier.

Tian, Y., Kanade, T., & Cohn, J. (2003). Facial expression analysis. In Li, S. Z., & Jain, A. K. (Eds.), *Handbook of face recognition*.

Tsutsui, T., Saeyor, S., & Ishizuka, M. (2000). *MPML: A multimodal presentation markup language with character agent control functions* (pp. 537–543). Proc. of WebNet.

Turunen, M. (2004). *Jaspis - a spoken dialogue architecture and its applications*. Ph.D. Dissertation, University of Tampere, Department of Computer Sciences A-2004-2.

Van de Burgt, S. P., Andernach, T., Kloosterman, H., Bos, R., & Nijholt, A. (1996). Building dialogue systems that sell. *Proc. NLP and Industrial Applications*, (pp. 41-46).

Wahlster, W. (2001). SmartKom: Multimodal dialogues with mobile Web users. *Proc. of International Cyber Assist Symposium*, (pp. 33-40).

Wahlster, W. (2003). *SmartKom: Symmetric multimodality in an adaptive and reusable dialogue shell* (pp. 47–62). Proc. of Human Computer Interaction.

Wahlster, W. (Ed.). (2006). *SmartKom: Foundations of multimodal dialogue systems*. Springer. doi:10.1007/3-540-36678-4

Walker, J. H., Sproull, L., & Subramami, R. (1994). Using a human face in an interface. *Proc. of SIGCHI conference on human factors in computing systems*, (pp. 85-91).

Walker, M. A., Litman, D. J., Kamm, C. A., & Abella, A. (1998). Evaluating spoken dialogue agents with PARADISE: Two case studies. *Computer Speech & Language*, *12*, 317–347. doi:10.1006/csla.1998.0110

Walker, M. A., Passonneau, R., & Boland, J. E. (2001). Quantitative and qualitative evaluation of Darpa Communicator spoken dialogue systems. *Proc. of 39th Annual Meeting of ACL*, (pp. 515-522).

Walker, W., Lamere, P., & Kwok., P. (2002). *FreeTTS: A performance case study*. Sun Microsystems, Inc.

Walsh, P., & Meade, J. (2003). Speech enabled e-learning for adult literacy tutoring. *Proc. of ICALT*, (pp. 17-21).

Ward, W., & Issar, S. (1994). Recent improvements in the CMU spoken language understanding system. *Proc. of ACL Workshop on Human Language Technology* (pp. 213-216).

Wasinger, R., Stahl, C., & Krüger, A. (2003). *Robust speech interaction in a mobile environment through the use of multiple and different media types* (pp. 1049–1052). Proc. of Eurospeech.

Webb, N., Benyon, D., Bradley, R., Hansen, P., & Mival, O. (2010). Wizard of Oz experiments for a companion dialogue system: Eliciting companionable conversation. *Proc. of International Conference on Language Resources and Evaluation (LREC 2010)*.

Wei, X., & Rudnicky, A. (2000). Task-based dialogue management using an agenda. *Proceedings of ANLP/NAACL Workshop on Conversational Systems*, (pp. 42-47).

Williams, J., & Young, S. (2007). Partially observable Markov decision processes for spoken dialogue systems. *Computer Speech & Language*, *21*(2), 393–422. doi:10.1016/j.csl.2006.06.008

Young, S., Adda-Decker, M., Aubert, X., Dugast, C., Gauvain, J., & Kershaw, D. (1997). Multilingual large vocabulary speech recognition: The European SQALE project. *Computer Speech & Language*, *11*, 73–89. doi:10.1006/csla.1996.0023

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., & Woodland, P. (2000). *The HTK book*. Microsoft Corporation.

Zapata, C. M., & Carmona, N. (2007). El experimento Mago de Oz y sus aplicaciones: una mirada retrospectiva. *Dyna rev.fac.nac.minas, 74*(151), 125-135.

Zhang, T., Hasegawa-Johnson, M., & Levinson, S. A. (2005). Hybrid model for spontaneous speech understanding. *Proc. of AAAI Workshop on Spoken Language Understanding*, (pp. 60-67).

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., & Hetherington, L. (2000). JUPITER: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, *8*(1), 85–96. doi:10.1109/89.817460

Zukerman, I., & Litman, D. (2001). Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, *11*, 129–158. doi:10.1023/A:1011174108613

KEY TERMS AND DEFINITIONS

Automatic Speech Recognition (ASR): Technique to determine the word sequence in a speech signal. To do this, this technology first detects basic units in the signal, e.g. phonemes, which are then combined to determine words. Some kind of grammatical information is used to determine more precisely the word sequence, by considering that some words are more likely than other taken into account the previous words in the sequence.

Dialogue Management (DM): Implementation of the "intelligent" behaviour of the conversational agent. It receives some sort of internal representation obtained from the user input and decides the next action the system must carry out. Typical actions are: i) to query the database module, ii) to generate a prompt to ask the user for additional data, iii) to generate a confirmation prompt to confirm unreliable data obtained from the user, iv) to provide help to the user, etc.

Embodied Conversational Agent (ECA): Humanlike computer-generated character that provides auditory and visual feedback for the user, which is particularly useful when the interaction takes place in noisy environments. Its complexity can vary significantly in terms of sophistication and complexity, from simple cartoon-like to complex animated human faces.

Fission of Multimodal Information: Opposite to the *fusion* operation, chooses the output to be produced through each output modality and coordinates the output across the modalities in order generate an agent's response appropriately for the user.

Fusion of Multimodal Information: Operation that combines the information chunks provided by the diverse input modules of the conversational agent in order to obtain a better understanding of the intention of the user. For example, the combination of acoustic information and visual information obtained from lip movements can enhance notably the performance of ASR systems, especially when processing low quality signals due to noise or other factors.

Galaxy Communicator: Distributed, message-based, hub-centred architecture to interconnect the main components of conversational agents. Among others, this architecture that has been used to set up the MIT's Voyager and Jupiter agents.

HADA-DS: Multimodal conversational agent in development to assist professors and students in some of their daily activities within a University's Faculty.

Natural Language Generation (NLG): Creation of messages in text mode, grammatical and semantically correct, which will be either displayed on screen or converted into speech by means of text-to-speech synthesis.

PARAdigm for DIalogue System Evaluation (PARADISE): One of the most employed proposals for globally evaluating the performance of spoken conversational agents. It combines different features in a single function that measures the performance of the agent in direct correlation with user satisfaction.

PROcedure for Multimodal Interactive System Evaluation (PROMISE): Extension to multimodality of the PARADISE framework. This procedure uses methods traditionally employed to evaluate spoken conversational agents, and specific methods to assess the characteristic properties of multimodal conversational agents.

R-Flow: Extensible XML-based architecture for the implementation of multimodal conversational agents, which is based on a recursive application of the Model-View-Controller design.

Speech Synthesis: Artificial generation of human-like speech. Currently, speech synthesis techniques can be classified into voice coding, parametric, formant-based and rule-based approaches. A particular kind of speech synthesis technique is called Text-To-Speech synthesis (TTS), the goal of which is to transform into speech of any input sentence in text format.

Spoken Language Understanding (SLU): Technique to obtain the semantic content of the sequence of words provided by the ASR module. It must face a variety of phenomena, for example, ellipsis, anaphora and ungrammatical structures typical of spontaneous speech.

VoiceXML: Standard XML-based language to access web applications by means of speech.

Wizard of Oz (WOz): Technique that uses a human called *Wizard* to play the role of the computer in a human-computer interaction. The users are made to believe that they interact with a computer but actually they interact with the Wizard.

XHTML+Voice (X+V): XML-based language that combines traditional web access using XHTML and speech-based access to web pages using VoiceXML.

ENDNOTES

[1] http://www.w3.org/TR/voicexml20/
[2] http://www.opoera.com/
[3] http://www.w3.org/TR/voicexml20/