Universal Access in the Information Society manuscript No.
(will be inserted by the editor)

Dialogue Based Interfaces for Universal Access

Christian Huyck

Received: date / Accepted: date

Abstract Conversation provides an excellent means of communication for almost all people. Consequently, a conversational interface is an excellent mechanism for allowing people to interact with systems. Conversational systems are an active research area, but a wide range of systems can be developed with current technology. More sophisticated interfaces can take considerable effort, but simple interfaces can be developed quite rapidly. This paper gives an introduction to the current state of the art of conversational systems and interfaces. It describes a methodology for developing conversational interfaces and gives an example of an interface for a state benefits web site. The paper discusses how this interface could improve access for a wide range of people, and how further development of this interface would allow a larger range of people to use the system and give them more functionality.

Keywords Dialogue, Universal Access, Conversational System, Conversational Interface

1 Introduction

Dialogue is a mode of communication that is efficient, highly effective, cross modal and flexible. Dialogue, or a conversation, can be enjoyable and enables the participants to fully explore problems and answers to those problems. Dialogue can work in very impoverished environments; for example, a telephone conversation has no visual interaction, and even the voice signal is quite poor; other conversations are held by simple interchanges of text as in internet chat.

Almost all people can participate in and communicate via dialogue. Sight is not needed; voice is not needed since communication can be by text; people with relatively severe memory limitations can converse; mobility is not a limitation. In only extreme

Christian R. Huyck
Middlesex University
Tel.: +44-208-411-5412
Fax: +44-208-411-5412
E-mail: c.huyck@mdx.ac.uk

circumstances are people unable to communicate via dialogue. Even extremely impaired people can communicate, all be it with difficulty, via eye movements.

Since dialogue is so effective, why do so few information systems rely on interfaces based on dialogue? The answer is simple. It is hard to develop relatively sophisticated dialogue based interfaces.

This paper will first introduce conversational systems, and give an idea of the current state of the art in dialogue interfaces. Next it proposes a system development methodology; this is not particularly novel but should give the reader an idea of the process of developing a conversational interface. An example development plan for an interface for a state benefits web site is then explored. Finally the paper discusses the benefits of a conversational interface and concludes.

2 Background

Computational dialogue systems have been developed for many decades. One of the most well known dialogue systems is Eliza [Weizenbaum(1966)]. The system interacts with the user via text. The user types in text, Eliza interprets it, and responds in text. Eliza poses as a psycho-analyst and it has been said that people confuse Eliza for a human.

Eliza uses a simple heuristic to make the user believe it is human. Eliza directs the conversation [Wilks(1996)] by asking questions and rarely answering questions. This allows Eliza to use a very simple pattern matching mechanism to understand the users input. For instance, if the user types in a sentence that contains a word that is a family member, like *mother*, Eliza understands the sentence to be about a family member. It then makes one of a few possible responses including that item; for instance it might say *Tell me more about your mother*.

While Eliza is interesting, it is not very sophisticated, and so the technology that it uses is only appropriate for very simple dialogues. At the other extreme, the standard test for an artificial system being intelligent is a dialogue.

The Turing test [Turing(1950)] is widely agreed to be a reasonable test for machine intelligence. To date, no system has come close to passing the test. In brief the test is conducted by a judge and two subjects: a machine and a person. All three participants are placed in separate rooms, and the judge communicates with each subject independently via a teletype. In the modern world the teletype could be replaced by internet chat. The judge then has a conversation with both subjects. If the judge can not tell which subject is human, or guesses at chance, the machine is intelligent.

The key to the Turing test is domain generality. Humans can talk about a wide range of subjects. No human can talk in depth on all subjects, but most pairs of humans can find something to talk about; that is they share expertise in some domain. If a machine had a similar range of topics that it could talk about, it could reasonably be called intelligent.

As the Turing test has not been passed, it can reasonably be concluded that no state of the art AI system is domain general. Indeed, this is the case. However, a wide array of systems function at or near human level performance in domain specific areas.

Indeed, there are conversational systems that function at or near human level performance in specific domains. For example, the TRAINS project [Allen et al(1995)Allen, Schubert, Ferguson, Heeman, Hwang, Kato Heeman and Allen(1995)] is a dialogue based system that allows a human to schedule trains in cooperation with a system. The communication medium, the interface, is a

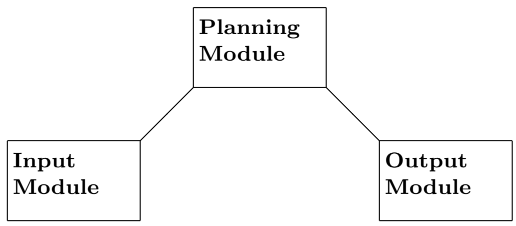

Fig. 1 Overall Structure of a Complex Dialogue System including Input, Output and Planning Modules

dialogue. The user is an expert at train scheduling that does not need any expertise in using a computer.

A standard format of this system is described in figure 1. The standard boxes describe basic modules. The Input box refers to a system that process the users input. This is typically called Natural Language Understanding; this is typically a sentence by sentence processor, though in this case it may keep a richer context for the dialogue. The Output box refers to the system that creates the dialogue that is sent to the user. This is typically called Natural Language Generation; for dialogue systems this may also retain rich dialogue structures. To some extent these are obvious. What is less expected is the planning module. This module is what really does things. In essence this is the underlying system, while input and output provide the conversational backbone. The planning module is domain specific. The input and output modules have domain specific features, but can be more general as language is largely domain general. The TRAINS system and other systems developed by this group use a planning module as the underlying system. For a different task such as document retrieval a different type of module would be useful.

Currently, this is the most sophisticated type of computer based dialogue system. There are, of course, simpler dialogue systems. For example, automated phone systems where the user (caller) responds to questions by pressing one of several options are dialogue systems. They are very simple and usually implement simple decision trees. Frequently, the last step of these systems is to transfer the call to a human operator to complete the conversation. Directed-dialogue systems require the user to answer a set of questions, and the questions can be answered via speech [Billi et al(1998)Billi, Canavesio, and Rullent]; indeed, the whole conversation happens on the phone.

There are a range of dialogue systems from simple press-number systems up to and beyond the TRAINS system. The goal of AI is that these systems will eventually become complex enough to pass the Turing test.

While dialogue systems remain an open research topic, dialogue systems are being developed for relatively complex domains right now. One type of system is for interactive question answering [Webb and Webber(2008)]. These systems are information retrieval systems and much of the research in this area makes use of the long history of information retrieval research. Most people use information retrieval systems such as keyword based search engines like Google. Interactive question answering systems extend the capabilities of keyword based systems by interacting with the user via conversation.

This conversation can be speech based or text based. A range of work has been done in this area including some on open domain systems [Quarteroni and Manandhar(2008)].

While dialogue can be text based, much current research is based on speech. Speech has the difficulty of being transient; both speaker and hearer have a constantly changing stream of input. A range of research explores how best to present speech, to restate when necessary, and confirm key parts of the dialogue. One sample of research deals with the special nature of interaction with older adults [Zajicek et al(2004)Zajicek, Wales, and Lee]; this notes that older users find it difficult to remember long instruction, even more so than younger users. So it is best to give little hints while the user is following instructions. There are a range of usability concerns [Dybkjaer and Bernsen(2000)] from the quality of speech production to providing adequate feedback.

There are also a vast number of chatbots or conversation bots [Goh et al(2008)Goh, Fung, and Depickere]. The Turing test is being run annually as the Loebner prize [Loebner(2009)]. All of the participants in this are conversational agents. These systems have no particular purpose save to convince the user that they know what they are talking about. Other systems have a particular purpose, such as answering queries on potential pandemics [Goh et al(2008)Goh, Fung, and Depickere].

Perhaps the most advanced dialogue work is on tutorial systems. In some cases, there is firm evidence that these systems are superior to reading or lectures [VanLehn et al(2007)VanLehn, Graesser, TannerJackson, The tutorial domain has the advantage of being able to direct the conversation, as the tutor is often in charge. The corresponding down-side is that the tutorial agent needs to determine what the student knows.

This is just a brief tour of some of the research in dialogue and some of the systems with dialogue based interfaces. Other research topics include (but are by no means limited to) user modelling [Jokinen(2006)], speech recognition, response planning [Seneff(2002)], and trust generation and support [Cassell and Bickmore(2003)].

Systems with dialogue interfaces are not simple to implement, but can be used to allow a wide range of users to access systems. Frameworks do exist for rapidly developing prototype conversational systems [Glass et al(2004)Glass, Weinstein, Cyphers, Polifroni, Chung, and Nakano]. Frameworks for speech based conversation are also available [Bohus and Rudnicky(2008)]. These frameworks and accompanying software can radically reduce the effort needed to develop dialogue based interfaces.

Dialogue systems are capable of allowing near Universal Access. Almost every human is capable of a conversation. Consequently, a conversational system can be developed to allow almost every human to interact with almost every system. While the development of such an interface may not be economically viable for all systems, it is possible.

Next, the paper proposes a methodology for developing dialogue interfaces to systems. This is certainly not the only methodology, but it is hoped that this will give the reader a better understanding of how a system could be developed and when it is within the range of current technology.

3 System Development Methodology

Standard system development methodologies apply. A feasibility study is useful to see if it is possible and economically viable to develop such a system. User groups should be consulted to, among other things, see the range of options available for the interface. Using existing tools, prototypes can be developed. These can then be given to user

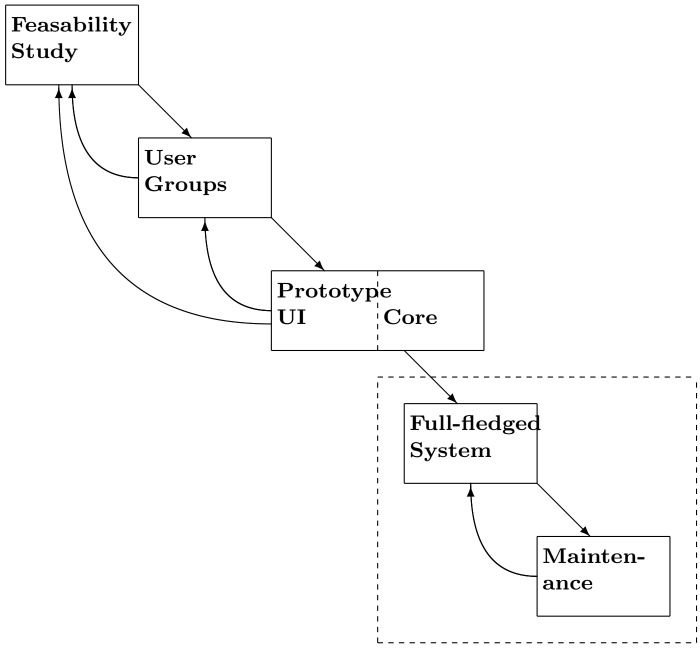

Fig. 2 A Simple Methodology for Building a System with a Dialogue Based Interface. The dashed box indicates a standard methodology.

groups to see problems and ways forward. Finally, the system needs to be maintained, particularly if the underlying data changes.

Figure 2 gives a pictorial representation of this method. Arrows denote time. It combines a dynamic prototyping software design methodology with a standard waterfall methodology. Initially, a system is developed through a series of prototypes, while the developers interact with user groups. At any step in the process, system development can return to any of the earlier phases. Once a prototype has met with approval of the user community, a standard design methodology can be used to roll out a solid product with a solid conversational interface.

3.1 Feasibility Study

Traditional system design incorporates requirements gathering and a feasibility study. The feasibility study is particularly important to dialogue systems. If the underlying system is infeasible, the overall system is obviously infeasible. Moreover with dialogue systems, the domain must be relatively simple to enable the conversational support that is needed for input and output modules.

For instance, it might be feasible to develop a chess playing system as an underlying module, but would it be feasible to develop a conversational system to interact with that module. The chess domain is relatively restricted, so it should be possible to develop a conversational interface.

As a second example, it is possible to develop a search engine to search for information on the internet. However, it might not be possible to develop a conversational system to support this system. The domain is more or less open as someone could ask for a document on any topic. Note that interactive question answering systems [Quarteroni and Manandhar(2008)] work in an open domain, but only answer simple questions. While impressive, this does not mean all tasks on open domains, such as document retrieval, are currently viable.

At this stage, it would be useful to decide the level of sophistication of the conversational system. Complex open ended dialogue systems can be developed but are much more complex than push-button and directed-dialogue systems.

3.2 User Groups

Another important process is to test user groups. This process should begin during the feasibility study and continue throughout the cyclic development cycle. Ideally this would involve a broad range of potential users.

In the formative stage of development, users can be brought into focus groups to see how they might interact with a system to use it to solve problems or do other activities. This can lead to specifications that explicitly include users' needs.

Wizard of Oz experiments can be developed as a form of early prototype. Users can interact with a human as if it were the system. If this is a text mode conversational system, a human can play the part of the system. In this case, a simple chat program would be sufficient as the interface between the user and the person acting as the system.

Once prototypes have been developed, they can be given to user groups. The users can interact with the systems to solve particular tasks, or merely use them for what they feel they need to use them for. This gives solid feed-back for the system developers. Bugs can be discovered, and new features requested. These interactions also provide a corpus. This corpus is particularly important for the development of modern Natural Language Processing systems because they often use machine learning techniques to improve their capabilities.

3.3 Prototyping

If the feasibility study is passed, and the user groups have found the proposed system promising, the next stage is prototype development. While each system is unique, a range of modern Natural Language Processing modules exist. Particularly early prototypes can take advantage of these modules.

For example, the General Architecture for Text Engineering (GATE) [Cunningham et al(1996)Cunningham, Wilks, and Gaizaus is a platform for combining existing modules. Modules that have already been integrated into the framework include lexical analyzers for part of speech tagging, gazeteers for proper name recognition, and parsers [Huyck(2000)] for getting the meaning of sentences from words, among a wide range of other systems. Moreover, these modules incorporate support for a range of different natural languages (e.g. English, French, and Chinese). These systems can be combined to enable the rapid development of a system that begins to understand language. GATE is an example framework. Other frameworks exist. The SpeechBuilder framework [Glass et al(2004)Glass, Weinstein, Cyphers, Polifroni, Chung, and Nakano]

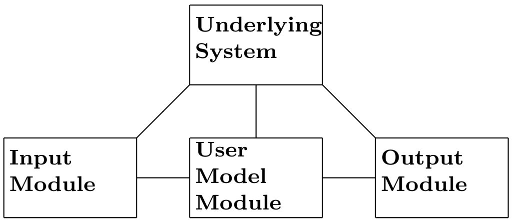

Fig. 3 Structure of a Dialogue System including User Model

allows rapid prototyping of dialogue systems using speech. Galaxy-II [Seneff et al(1998)Seneff, Hurley, Lau, Pao, Schmid, and Zue] provides a client-server architecture for accessing information by speech dialogues.

Similar systems can be combined to develop an output module. Moreover it is relatively easy to create output systems that simply pre-package the output. This approach lacks flexibility, but takes advantage of the skills that almost all humans have in language interpretation.

Later prototypes can improve on these modules making them more flexible. They can also incorporate a better understanding of the domain that has been gained from the creation of these earlier prototypes and the analysis of user interactions from user groups (see section 3.2).

In figure 2, the prototypes are broken into UI and core. These components may be developed independently, or together. Either way they should be tested together, and shown to user groups together.

Another option is to include a user model. While user modelling is an active research area [Komatani et al(2003)Komatani, Ueno, Kawahara, and Okuno, Jokinen(2006)], simple models can add a great deal of flexibility to a system. A user model would lead to a more sophisticated system. Figure 3 shows this system. This extra module would facilitate extra support for each unique user. This would be an ideal place to retain information about the particular conversation that is taking place. It would also be an excellent place to store information about atypical properties a particular user might have. For instance, if a user has a short-term memory limitation, this module could be used to give the user extra support for retaining past information about the conversation and support complex goal pursuit.

3.3.1 The Underlying System

This is the most domain specific portion of the system. It is also the subsystem that does the real work. In the TRAINS system, this is the planning system that schedules the trains.

Little can be said about this module because it is going to be specific to each task. It is plausible that an existing system already exists, and all that is needed is to "plug-in" the interface.

Many systems for database retrieval have been built with SpeechBuilder [Glass et al(2004)Glass, Weinstein, Cyphers, Polifroni, C Information retrieval in selected domains (e.g. restaurant guide and weather information [Seneff et al(1998)Seneff, Hurley, Lau, Pao, Schmid, and Zue]) has been implemented. Complex problems like train scheduling [Allen et al(1995)Allen, Schubert, Ferguson, Heeman, Hwang, Kato, Light, Martin can be very complex to develop.

3.3.2 Interface Modalities

Conversations are cross modal. That is, conversations happen via text, via speech, via vision and even via touch. Textual conversations happen when the communication medium is text, as in a postal conversation, an email conversation, or internet chat. The traditional modality is speech; participants take turns speaking and listening. Sign language is a way of having a visual conversation. Finally, text can be automatically translated to Braille to enable a tactile conversation.

It is possible for a system to communicate in all of these modalities. Additionally, extra information such as gesture or pictures may be available. Virtual environments can be used to support the generation of facial expressions and gesture [Baldassarri. et al(2008)Baldassarri., Cerezo, and Seron], but these will certainly complicate the interface.

Deciding which modalities to support is a key decision in developing the system. Early prototypes might support one modality, say text, with later modality additions for later prototypes. However, thinking ahead may enable better support of extra modes.

Another issue is the range of languages that a system supports. One obvious question is does it support multiple languages such as Urdu and English. It should be relatively easy to modify a conversational system to account for multiple languages, but each causes an extra burden.

A related question is the sophistication of the language that is used. One user might be comfortable with relatively advanced word use, while another requires simple words. Dialect is also an issue, and accent is an issue for speech-based systems.

Again, early prototypes can incorporate one language, with later prototypes adding others, other dialects, or flexible usage. However, it is useful to plan before hand.

3.4 Repeat

One key to development is to repeat the process. Prototype systems are useful for user groups, and even end users. These can be rolled out and evaluated, and future prototypes can take into account the successes and failures of these earlier systems.

4 Sample Development Plan

This section will describe a relatively complex conversational interface. The interface has not been implemented, and has only been planned in the scantest of ways, yet an initial development plan is presented. The interface is for a benefits system for the UK's Department of Work and Pensions (DWP).

The existing web site http://www.dwp.gov.uk/ has several thousand pages, and supports several hundred forms. It changes quite frequently; the recent addition of the 252 kilobyte pdf file entitled “The Pension Protection Fund (Miscellaneous Amendments) Regulations 2008” illustrates the scale of the change problem. As anyone who has tried to fill in a tax form knows, finding the proper form and filling it in properly is often difficult for even the most educated.

This site supports a wide range of people including some of the most vulnerable people in the UK. For instance, someone who has recently been incapacitated on the job will need to fill in forms from this site to receive the benefits to which they are entitled.

4.1 Feasibility Study

Would it be useful to build a conversational system for the DWP, and is it viable? If viable, what type of system is viable?

The existing interface has a reasonable hypertext set up. The search engine is not very useful; for example, typing in incapacity benefit gives 5289 hits though the first seems like a reasonable page describing incapacity benefits. So, there is room for improvement.

A directed-dialogue approach would be simple, and would probably be useful to help people find particular information, and should also be useful to help them fill in forms. A more sophisticated conversational system might also be useful. It would enable more sophisticated retrieval and form filling. Developing an open-ended conversational system could be very expensive, so a prototyping approach will be used. Each prototype will be evaluated before the next is designed and developed. This will allow a gradual approach.

4.2 User Groups

User groups will be needed for design input, requirements specification, and prototype testing. A wide range of people should be included in the groups and should obviously include a cross section of users. Perhaps more subtly, people working in the organisation should be included. This will help set priorities, and will provide developers with access to expertise. If more senior members are included, the cost to benefit may be optimised as these people will be able to direct the system to support the most needy users and the most used portions of the network.

The user groups should contain people from a wide range of educational and linguistic backgrounds as the users of the system will be diverse. Hopefully, picking users from existing users will address this issue, but known properties about the range of users should be considered. For example, Urdu speakers may be less willing to join the user group, but make up a significant portion of the actual user population. In this case, extra Urdu speakers should be recruited for the group.

Similarly people with a wide range of functionality should be included. If it may be possible to develop an interface for a person, they should at least be considered for the user group. For instance, conversational systems work well for people with visual impairments, so some should be included in the user group. As the DWP site will be used by people on disability benefit, people with a range of disabilities should be included in the user group. The early prototypes may not be able to allow access to all, even the final system may not allow access to all, but placing as wide a range of users as possible will encourage the developers to consider the possibility of providing interface support for them.

4.3 System

The system will follow the subsystem model of figure 3. There will be Natural Language Understanding and Natural Language Generation subsystems.

The underlying system is initially undetermined but will be an aid to information retrieval and to form filling. It needs to be tied closely to the existing web site.

The user model is also an important component. Security is a concern so the user may not want to enter any personal details. Even given this relatively strict limitation, the system can act as a guide to help the user find information. The user model can be merely a session long model that lends support for the conversation. With fewer security concerns, the user model may allow for special support for particular individuals and may retain information on their status and past conversations.

4.4 Repeat

The proposal is to develop three prototypes of increasing capability and sophistication. This will allow for ongoing testing, and early use. It allows for flexible planning, including stopping early if necessary. It also allows for data to train later prototypes.

It is proposed that the initial prototype is relatively simple. It will accept text input, and produce a relatively restricted range of textual outputs. It will produce simple English to allow weaker English speakers to understand the conversation. A subset of the documents will be used including the most commonly used documents, and the most important documents. A hook to the web pages will be provided to take advantage of the existing web infrastructure. Privacy issues will be explored particularly in conjunction with the user profile. An existing framework like GATE [Cunningham et al(1996)Cunningham, Wilks, and Gaizauskas] will be used to build the system.

If the system is to be developed, this proposal should be taken to a user group or user groups. An example Wizard of Oz system will be used to test the basic ideas of the prototype before they are actually implemented. This process should develop particular scenarios that the system must address.

The prototype must be evaluated. This should include user groups, and groups from the DWP. It must be checked on the scenarios that have been explicitly chosen, and it must be tested on other scenarios. It may not succeed on these, but if it fails it should do so gracefully.

At this point the second prototype should be specified and designed. However, it is proposed that the extension to the prototype might include the following extensions. The second prototype should support both speech and text communication. An extra language can be included (e.g. Urdu). The document set that is supported should be expanded. The old set of documents should continue to be supported; as this will most likely contain modifications, a plan for supporting the changing web site should be devised and at least partially implemented in this prototype. Again, this prototype release should be followed by a period of evaluation.

Finally, a third prototype might include further extensions. It could include a third language (e.g. Polish) or more. It would include further documents and might even include support for the full set of documents; it would need support for the changing document set. It could included further support for people with memory deficits and extra user model support for the visually impaired. Of course, this would be followed by a period of evaluation.

5 Discussion

Conversational interfaces can range from rather simple to highly sophisticated. Conversation is good for many tasks for everyone and is the natural way humans exchange information. It can enable the conversants to focus on specific problem areas.

Support is needed for conversation that is somewhat atypical for user interfaces. The "what were we talking about" phenomenon [Gruenstein and Cavedon(2004),Weng et al(2004)Weng, Cavedon, Raghunathan, Cheng is common in conversations. This can be addressed by the conversational system keeping track of goals or assumed goals. It can then prompt the user or direct the conversation back to a goal.

Conversational interfaces can allow people with a range of sensory deficits to use the system productively. Visual and auditory impairments can be easily overcome with conversational interfaces. To a great extent, mobility impairments can also be overcome.

Perhaps the most obvious benefit is for people with visual impairments. No windows are required; indeed no visual output at all is needed. Output is readily available in text and speech.

Support needed for a typical user should not need to be significantly modified for the visually impaired. The fully-sighted user might be able to easily see a trace of the recent portion of the conversation - via, for instance, the chat window - and this could be made easily available to a visually impaired user by a backward scrolling technique [Hurst et al(2005)Hurst, Burfent, and Gotz].

Conversation is also useful for people with relatively minor visual impairments. Things as minor as eye strain could be reduced.

Most existing interfaces do not use sound, so auditory impairments are less of an issue. None the less, conversation, via text, is available for people with auditory impairments.

Another issue is people with mobility impairments. If someone can not use or has difficulty with a mouse or a keyboard, then interacting with a computer can be particularly difficult. If that person is capable of audio conversation, a conversational interface should be as easy for them as for the typical user. Even people with significant mobility impairment and audio impairment should be able to converse. In this case, a picking word interface can be particularly useful.

There are other types of impairments aside from sensory impairments. Many memory impairments can be addressed by conversational systems. If a person has frontal lobe damage and has difficulty maintaining goals, a conversational system can focus the person on a goal. If a person has short-term memory limitations, a conversational system can present information again to facilitate understanding.

Conversation can also provide support for users that are typically unable to use computers. Those with poor computer or typing skills can interact via speech. Those with poor writing and reading skills, can also interact via speech. As a visual record of a conversation is useful and those with poor reading skills or visual impairments do not have access to this visual record, extra support can be provided in the form of conversational reminders.

Conversational interfaces can be built for any natural language. While it may not be feasible to build a conversational interface for a particular system for all languages, support for multiple languages is possible. If there are enough users who speak a particular language, support can be developed for them.

6 Conclusion

Conversational systems can be built with current technology. These offer a host of benefits to allow Universal Access including the ability to be used by people with sensory impairments, memory impairments, those with limited computer skills, and those from a range of linguistic backgrounds.

While conversational systems offer these benefits, they come at a cost. Conversational systems are still an active research area and sophisticated conversational interfaces may be very expensive to build. Fortunately, simple dialogue based interfaces can be developed relatively simply. Consequently, it is important to select the right level of sophistication. It is much more simple, and thus less expensive, to build a directed-dialogue interface than to build an open ended dialogue interface. On the other hand, complex dialogue based systems can be developed.

Conversational systems potentially solve a host of interface problems. When deciding on an interface, conversational systems should be considered.

Acknowledgements This work was supported by EPSRC grant EP/D059720 - Natural Language Parsing using Cell Assemblies: Computational Linguistics with Attractor Nets.

References

[Allen et al(1995)Allen, Schubert, Ferguson, Heeman, Hwang, Kato, Light, Martin, Miller, Poesio, and Traum] Allen J, Schubert L, Ferguson G, Heeman P, Hwang C, Kato T, Light M, Martin N, Miller B, Poesio M, Traum D (1995) The TRAINS project: A case study in building a conversational planning agent. Journal of Experimental and Theoretical AI 7:7–48

[Baldassarri. et al(2008)Baldassarri., Cerezo, and Seron] Baldassarri S, Cerezo E, Seron F (2008) Maxine: A platform for embodied animated agents. Computers and Graphics 32:430–437

[Billi et al(1998)Billi, Canavesio, and Rullent] Billi R, Canavesio F, Rullent C (1998) Automation of telecom italia directory assistance service: Field trial results. In: Proc. IVTTA, pp 11–16

[Bohus and Rudnicky(2008)] Bohus D, Rudnicky A (2008) The RavenClaw dialog management framework: Architecture and systems. Computer Speech and Language

[Cassell and Bickmore(2003)] Cassell J, Bickmore T (2003) Negotiated collusion: Modelling social language and its relationship effect in intelligent agents. User Modeling and User-Adapted Interaction 13:1:89–132

[Cunningham et al(1996)Cunningham, Wilks, and Gaizauskas] Cunningham H, Wilks Y, Gaizauskas R (1996) GATE: A general architecture for text engineering. Computational Linguistics

[Dybkjaer and Bernsen(2000)] Dybkjaer L, Bernsen N (2000) Usability issues in spoken language dialogue systems. Natural Language Engineering 6:3:243–271

[Glass et al(2004)Glass, Weinstein, Cyphers, Polifroni, Chung, and Nakano] Glass J, Weinstein E, Cyphers S, Polifroni J, Chung G, Nakano N (2004) A framework for developing conversational user interfaces. In: Proceedings of CADUI

[Goh et al(2008)Goh, Fung, and Depickere] Goh O, Fung C, Depickere A (2008) Domain knowledge query conversation bots in instant messaging (IM). Knowledge-Based Systems 21:681–691

[Gruenstein and Cavedon(2004)] Gruenstein A, Cavedon L (2004) Using an activity model to address issues in task-oriented dialogue interaction over extended periods. In: AAAI Spring Symposium on Interaction beween Humans and Autonomous Systems over Extended Operation

[Heeman and Allen(1995)] Heeman P, Allen J (1995) The TRAINS 93 diaolgues. Tech. rep., Univ. of Rochester

[Hurst et al(2005)Hurst, Burfent, and Gotz] Hurst W, Burfent C, Gotz G (2005) Forward and backward speech skimming with the elastic audio slider. In: Proceedings of HCI 2005

[Huyck(2000)] Huyck C (2000) A practical system for human-like parsing. In: Horn W (ed) Proceedings of the 14th European Conference on Artificial Intelligence

[Jokinen(2006)] Jokinen K (2006) Adaptation and user expertise modelling in AthosMail. Universal Access Information Society 4:374–392

[Komatani et al(2003)Komatani, Ueno, Kawahara, and Okuno] Komatani K, Ueno S, Kawahara T, Okuno H (2003) Flexible guidance generation using user model in spoken dialogue. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp 256–263

[Loebner(2009)] Loebner H (2009) Home page of the loebner prize. Tech. rep., URL

[Quarteroni and Manandhar(2008)] Quarteroni S, Manandhar S (2008) Designing an interactive open-domain question anwering system. Natural Language Engineering 15:73–95

[Seneff(2002)] Seneff S (2002) Response planning and generation in the *mercury* flight reservation system. Computer Speech and Language 16:283–312

[Seneff et al(1998)Seneff, Hurley, Lau, Pao, Schmid, and Zue] Seneff S, Hurley E, Lau R, Pao C, Schmid P, Zue V (1998) Galaxy-ii: A reference architecture for conversational system development. In: Proc. ICSLP-98, vol 3, pp 931–934

[Turing(1950)] Turing A (1950) Computing machinery & intelligence. Mind 59:236:433–60

[VanLehn et al(2007)VanLehn, Graesser, TannerJackson, Jordan, Olney, and Rose] VanLehn K, Graesser A, TannerJackson G, Jordan P, Olney A, Rose C (2007) When are tutorial dialogues more effective than reading. Cognitive Science 31:1:3–62

[Webb and Webber(2008)] Webb N, Webber B (2008) Special issue on interactive question answering: Introduction. Natural Language Engineering 15:1:1–8

[Weizenbaum(1966)] Weizenbaum J (1966) Eliza - a computer program for the study of natural language communication between man and machine. Communications of the Association for Computing Machinery 9:36–45

[Weng et al(2004)Weng, Cavedon, Raghunathan, Cheng, Schmidt, Bratt, Mishra, Peters, Zhao, Upson, Shriberg, and Bergmann] Weng F, Cavedon L, Raghunathan B, Cheng DMH, Schmidt H, Bratt H, Mishra R, Peters S, Zhao L, Upson S, Shriberg E, Bergmann C (2004) A conversational dialogue system for cognitively overloaded users. In: Proceedings of the 8th International Conference on Spoken Language Processing

[Wilks(1996)] Wilks Y (1996) Natural language processing. Communications of the Association for Computing Machinery 39:11:60–62

[Zajicek et al(2004)Zajicek, Wales, and Lee] Zajicek M, Wales R, Lee A (2004) Speech interaction for older adults. Universal Access Information Society 3:122–130