

# From Question Answering to Spoken Dialogue: Towards an Information Search Assistant for Interactive Multimodal Information Extraction

*Rieks op den Akker, Harry Bunt,  
Simon Keizer, Boris van Schooten*

Faculty of Arts, Tilburg University, Tilburg, Netherlands  
Faculty EWI, Universiteit Twente, Enschede, Netherlands

infrieks@cs.utwente.nl, harry.bunt@uvt.nl,  
s.keizer@uvt.nl, schooten@cs.utwente.nl

## Abstract

This paper gives an overview of issues related to extending simple question answering (QA) with dialogue capabilities, when designing a multimodal interactive information extraction system for a large, though restricted, domain.

We present the way in which these issues are approached in the IMIX program. The IMIX demonstrator system, under development in this program, may be considered the most difficult case of QA, answering non-factoid questions in a large domain, and accepting speech input as well. We describe our approach to the addition of dialogue capabilities to this system.

We will look at QA from a dialogue system perspective and from a HCI perspective, and consider the consequences of our choice of interaction metaphor, the ‘information search assistant’.

## 1. Introduction

This paper presents ongoing research within the framework of the IMIX (Interactive Multimodal Information Extraction) program, a Dutch national multiproject research effort concerning Dutch language and speech technology. IMIX brings together academic partners<sup>1</sup> and partners from industry<sup>2</sup> to collaborate on research involving question answering, speech recognition, speech and language generation, automatic ontology generation, and dialogue management. The collaboration involves both fundamental research and the development of a demonstrator system, in which the various technologies are combined into an interactive multimodal information extraction system for the domain of medical encyclopedic information.

The IMIX program now includes two projects concerned with dialogue management: VIDIAM (Dialogue Management and the Visual Channel) and PARADIME (Parallel Agent-based Dialogue Management Engine), that will cooperate in developing a dialogue manager for the demonstrator system.

Currently, a baseline system has been developed, providing question answering (QA) functionality without dialogue. The user can ask isolated, self-contained questions about the medical domain, using either speech or keyboard. The system answers with a matching document fragment in the document

---

<sup>1</sup>The universities of Groningen, Tilburg, Twente, Nijmegen and Amsterdam.

<sup>2</sup>Textkernel, providing support software, and Het Spectrum Publishers and MerckManual, providing electronic documents containing medical encyclopedic information.

database, which may be both spoken and displayed as text. Integrating a dialogue manager into the system will provide for assistance of users in finding information through an interactive and cooperative process. In the mixed-initiative dialogues that will be supported, users will additionally be able to ask the system for clarification, make corrections to the system’s interpretation of utterances and ask follow-up questions, and the system may additionally ask the user for clarification and ask verification questions. In addition to dialogue functionality, IMIX will also present pictures as part of the answers, and will enable the user to point to both the pictures and text on the screen as part of the dialogue.

To make clear what dialogue functionality the IMIX demonstrator will have, we choose the metaphor of an ‘Information Search Assistant’, which means that the system should have interactive capabilities in common with an assistant librarian who can provide help in finding answers to questions by identifying relevant documents and parts of documents.

The paper is organised as follows: in Section 2, we will discuss the dialogue capabilities that the IMIX demonstrator should have, taking into account (1) the requirements determined by the IMIX program as a whole, and the use of the Information Search Assistant in particular, and (2) the required dialogue functionality from a HCI perspective. In Section 3 we outline the approach to dialogue management that will be taken in the IMIX project, evolving from the requirements on the functionality of the system’s dialogue manager discussed in Section 2, and we indicate a partial system architecture to implement this functionality.

## 2. Dialogue Functionality for QA

The IMIX demonstrator is intended for casual users, i.e., users who have no professional knowledge of the medical domain, who use the system only occasionally, and who have not received any special training in using the system.

The addition of dialogue capabilities to a bare QA system is attractive for several reasons. First, it is well known that users are often unable to express their need for information in a single, self-contained question. This is especially so for casual users (as opposed to frequent, professional users) and complex information domains. Such users typically do not know precisely what can be asked, since they have no detailed knowledge of the information that is available. Moreover, they often have a desire for information which is not very articulate, especially when the information domain is relatively unknown. These circumstances make it desirable for the user to be able to not just

fire a self-contained question at the system, but to be able to interactively determine which information is wanted, and if this information is fairly complex (which is typically the case for medical information), to be able to interactively explore the various aspects of the information in a sequence of exchanges that forms a coherent dialogue.

## 2.1. Dialogue Functionality from a HCI Perspective

From a HCI perspective, the most important property of an interactive system is its *usability*, defined as the effectiveness, efficiency and satisfaction with which users can achieve specific tasks in a particular environment (ISO definition 9241).

According to HCI theory, the usability of the interface of an interactive system has three main components: learnability, flexibility, and robustness [1]. Learnability concerns all those aspects that make a system more easy to learn how to use. Flexibility refers to the multiplicity of ways user and system can exchange information, allowing the interaction to be organized in ways that suit the user. Robustness covers those properties that prevent the interaction from breaking down in problematic or unanticipated situations. In this section we translate the relevant properties determining the usability of the IMIX demonstrator system to requirements on the functionality of the system's Dialogue Manager (DM).

### 2.1.1. Learnability

Learning how a system can be used, means for the user that he forms a mental model of the system [2]. One way to support this process is by using an interaction metaphor, i.e. designing the interactive behaviour of the system in a way that mimics interactive behaviour of a kind that the user is already familiar with. The Information Search Assistant (ISA) metaphor that we have chosen is primarily an interaction metaphor. For the user this means that the system is comparable to an assistant who has little specific knowledge of the medical domain, but who is an expert on how to find information. When the user poses a question which is too vague or too poorly formulated to be answerable, the assistant helps in obtaining an information request that has a greater chance of being satisfiable.

The use of a human agent metaphor also implies that the DM should be able to handle spontaneous communication of the kind that casual users may use. This includes the various types of questions, informs, answers, and requests that are commonly distinguished in the context of information-seeking dialogues (see e.g. [3]). The system should also support the use of dialogue acts that are conventionally used to open and close a dialogue and to identify itself to the user.

### 2.1.2. Flexibility

An important flexibility property is that no artificial constraints are imposed on the user: he should not be forced to go through a number of interactive steps that he considers irrelevant, and he should be allowed to determine the direction of the dialogue. This implies that the user should have control over who has right to contribute to the dialogue at any moment; in other words, he should have *turn management acts* at his disposal. Moreover, the user should be allowed to open, close, and shift between topics at will, and prevent the system from moving to an undesired topic. This means that the Dialogue Manager should support the use of *topic management acts* by the user.

### 2.1.3. Robustness

The robustness of the interaction covers features which support the successful achievement and assessment of the user's goals. Central to this is the property of *observability*, which allows the user to have a good view of what the system is doing at all times. In terms of dialogue acts, observability is supported by the system providing adequate feedback about its processing of user inputs, and if the user is nonetheless uncertain about what is happening, he should be allowed to elicit additional feedback. In particular, the system should tell the user what it has understood, and what kind of query is submitted to the underlying QA engine. Another feature of robustness is *recoverability*, which is the support of actions to undo the effects of previous interaction if something went wrong. This functionality requires the DM to support the use of interaction management acts by the user, aimed at correcting actions performed by himself or by the system.

## 2.2. Dialogue Functionality from a Technical Perspective

Since there exist various QA systems with and without dialogue capabilities we will first compare IMIX with other QA systems by identifying different classes of QA:

1. **QA with structured DB.** While current QA system usually use unstructured data, older QA systems often used a structured DB (see for example [4]). The presence of a structured DB implies that the domain is very limited. Such systems are usually database query front-ends, and support only limited classes of questions.
2. **QA with semi-structured or unstructured DB and factoid questions with simple answers.** This involves questions like: *How many people suffer from RSI each year?* and *What does the abbreviation "RSI" stand for?*<sup>3</sup> For this class of QA, some relatively straightforward strategies have proven to be effective. In particular, a successful technique is to classify questions as 'date question', 'person question', 'location question', etc., and to use a specific answer finding strategy for each type of question.
3. **QA with semi-structured or unstructured DB and non-factoid questions.** The IMIX system belongs to this class. The questions posed by casual users will often require relatively long, explanatory answers. Such questions may be considered inherently ambiguous, as the kind and the amount of explanation required depends on the user's information need, thus taking us beyond factoid questions (cf. [5]). Examples of such questions are: *Why do people get RSI? What sort of exercises are there against RSI? What is the best way to cure RSI?* Answering non-factoid questions is notoriously difficult, and is not addressed by most QA systems. The addition of 'definitionoid questions' to factoid questions in TREC QA 2003 [6] was already considered a big step forward in functionality. We should note that in IMIX, life is made a little easier than in regular QA, because the precision requirement of the text fragment that is retrieved is relaxed with respect to typical factoid QA. Instead of one specific phrase that is the exact answer, IMIX retrieves a sentence or paragraph which contains the answer within it.

<sup>3</sup>RSI, Repetitive Stress Injury, is one of the subdomains of medical information that will be represented in the IMIX demonstrator.

Most interactive information dialogue systems are QA class 1 systems. Existing class 2 and class 3 QA dialogue systems typically have very limited dialogue capabilities.

The BirdQuest system [7] is an example of a class 2 QA system. It introduces an offline 'information extraction' step that automatically converts an encyclopedia into a structured DB. This assumes that the data is structurable into a structured DB, which implies again a small, limited domain. BirdQuest also supports some natural language dialogue. Its understanding of the domain is centred around the fields found in the DB, and does not go much beyond this.

An example of a class 3 QA system is HITIQA [5], which is based on document retrieval. It extracts topic information for each document segment, and matches this with the topic information from the question to determine how well each segment matches the question. In addition, it applies a similarity measure to retrieved documents to determine if they represent the same or very different answers. It can propose system-initiated narrowing or expansion of the document set, with the help of the topic distance measure. It also supports dialogue to some extent.

From a system requirements point of view, the IMIX demonstrator will need to feature at least the following dialogue functionalities that we find to some extent in some other class 2 and class 3 QA system.

- **User follow-up questions.** Instead of a single question, a user may pose a series of questions, referring to previous questions and answers in each question. Work in this area has been done for example by De Boni [8], and in the ASKA project [9]. Follow-up questions were also included in the TREC 10 (2001) competition as a "context questions" task [10], and the QAC 2 competition [11]. It was subsequently dropped from TREC because of methodological problems with the evaluation.
- **System clarification questions.** When the system cannot find an answer to a user question, the system may pose questions to the user to obtain sufficient information to make it succeed. This is represented for example in the ASKA [9], SPIQA [12], and HITIQA [5] projects. Basically, the system asks for extra information to widen or narrow the number of documents or answers retrieved. The approaches vary, for example, SPIQA asks for extra information on a particular term in the initial question, HITIQA asks for inclusion/exclusion of specific topics depending on the search result. This type of information transfer is however less common in the QA world. The QA competitions do not include any kind of system initiative.

For adequately supporting the non-expert user, the IMIX DM will have to support not only clarification questions by the system, but also clarification questions by the user, since the answers may contain terms that are not quite clear to the user.

Since the IMIX system is meant to be able to answer non-factoid questions, the system should also support certain forms of metacommunication to determine if the user is satisfied with the information provided as an answer. This means that the system should be able to respond adequately to user inputs that do not express a question, but do provide additional information for the system for finding more satisfactory answers.

While an unstructured DB and a large domain precludes the use of a 'deep' semantic model of user's information needs, the system can at least handle requests concerning the length and

number of answers, and deal with a user's evaluative feedback concerning a given answer. Also, it is useful for the system to give information about confidence and the source of the answer. Additionally, information from the available semantic tagging tools can be used to improve the user question's content interactively.

For IMIX, we consider the following as technically feasible additions as well: question reformulation requests by the system, clarification of words and content of images by the user, detection of out-of-domain questions, and browsing through dialogue history.

### 3. IMIX Approach

#### 3.1. Approach to Dialogue Management

The most common approach to dialogue management is the slot-filling approach. Here, the DM's main task is to help fill in a number of information slots that together define a user's query or command. The possible slots and queries are predetermined. Its main advantage is simplicity. QA class 1 dialogue systems most often use the slot-filling approach, which is appropriate for constructing queries to a structured DB. For QA class 2, slot filling might still be appropriate, as we may go some way in modelling questions by having one query type with one or more slots per question type.

For QA class 3, slot filling is less appropriate. What comprises a valid query can only be determined for each case using analysis of various sources of information: the output of the QA engine, and consultations of ontologies and NLP tools. Instead, we follow the information state approach, which is a more general approach. Here, dialogue behaviour is viewed in terms of dialogue acts, i.e. as communicative actions aimed at influencing the current information states of both participants, also known as the current 'dialogue context' (see e.g. [13]). The dialogue context includes the dialogue history up to the current point in the dialogue, as well as information about information needs that have been expressed and information that has been provided. In addition, the context model will include information about the processing of user inputs by the system and of processing of system outputs by the user, in order to be able to generate and understand communicative feedback.

Following an approach where dialogue acts and context modelling are main ingredients means that the DM will be organised around on the one hand the recognition of the user's dialogue acts, with their associated context updates, and on the other hand on the generation of dialogue acts as triggered by features of the current dialogue context. The implementation of these concepts in the IMIX DM will follow the agent-based architectural paradigm that has been successfully applied in several recent dialogue systems, and that is being developed for dialogue management also in the JASPI system (see [14]).

For its functioning, the DM will have to be able to do some inferential processing, for instance to apply domain knowledge in order to deal with a user question that is too broad or too narrow.

A dialogue system also needs intelligence about interaction. To achieve this, the DM requires a kind of ontology of interaction elements. A dialogue act taxonomy like the one developed in Dynamic Interpretation Theory (see [3]) will be used as a basis for the ontology to be used in the IMIX DM.

To illustrate the kind of dialogue phenomena we would like to support, an example dialogue is given below, containing user follow-up questions (U2 and U4), and a system clarification re-

quest (S2).

U1: welke spieren zijn betrokken bij RSI?  
(which muscles are involved in RSI?)  
S1: vooral de spieren in de armen, handen en de nek zijn betrokken bij RSI.  
(mostly the muscles in the arms, hands and neck are involved in RSI.)  
U2: o, dus niet die in de rug?  
(oh, so not those in the back?)  
S2: pardon, welke spieren zei u?  
(excuse me, which muscles did you say?)  
U3: de rugspieren.  
(the back muscles)  
S3: pijn in de bovenrug kan ook met RSI samenhangen.  
(pain in the upper back may also be related to RSI)  
U4: maar, eh, komt dat vaak voor?  
(but, erm, does that occur a lot?)  
S4: sorry, daar heb ik geen informatie over.  
(sorry, I have no information on that)

After the first question-answer pair concerning the domain, U gives feedback to S: first positive feedback on the level of interpretation (“o” implies that U has heard and understood the previous utterance) and then negative feedback on the level of evaluation, which can at the same time be seen as a new question about the domain. The clarification question in S2 has a negative feedback function on the level of perception. S4 does not give an answer to U4, but negative feedback on execution level (S is not able to give the answer).

### 3.2. System Architecture

The DM, the central component in a typical dialogue system, acts as a mediator between the user and the QA engine (see figure 1). The QA accepts both a question in textual form and a n-best list of speech recognition results, and replies with a list of answers, with a confidence score for each answer. Beside the confidence score, little meta information is passed between DM and QA. Still, DM and QA can share knowledge by means of shared resources, like a shared domain ontology and shared NLP tools like POS tagger and NL parser. The DM may pass additional question analysis information to the QA, along with the question. Note the analogy of the architecture to the ISA metaphor.

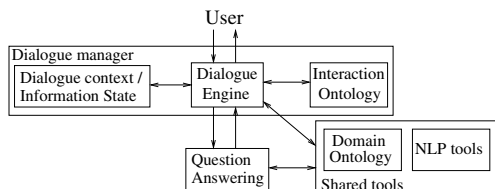


Figure 1: *IMIX architecture*. Access to the QA module is mediated by a dialogue manager. Dialogue manager and QA have no access to each other’s internals, but there are some shared resources (domain ontology and NLP tools).

## 4. Conclusions and future work

We presented the ISA metaphor as a basis for the user’s mental model. Ideally, we should verify the mental model of real users with that of our design, and evaluate its implications for user expectations of the system’s capabilities, and for interaction in general. However, full user evaluation of the system is not part of the project. Still, the system will be developed in several steps, in between which dialogues with in-field user groups will be collected. This dialogue data will be used for evaluation.

Future development steps are directed toward a system that allows users to refer in their speech or text input to non-textual items displayed by the system in response to the users informa-

tion request.

## Acknowledgement

The IMIX program is funded by the Netherlands Organisation for Scientific Research (NWO).

## 5. References

- [1] J. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction (3rd edition)*. Harlow: Pearson/Prentice Hall, 2004.
- [2] D. Norman, “Some observations on mental models,” in *Mental Models*, D. Gentner and A. L. Stevens, Eds. Lawrence Erlbaum, 1983, pp. 7–14.
- [3] H. Bunt, “A framework for dialogue act specification,” in *Fourth Joint ISO-ACL Workshop on the Representation of Multimodal Semantic Information*, 2005.
- [4] W. Bronnenberg, H. Bunt, J. Landsbergen, P. Medema, R. Scha, W. Schoenmakers, and E. v. Utteren, *The Question Answering System PHLIQAI*. McMillan, London, 1980.
- [5] S. Small, T. Liu, N. Shimizu, and T. Strzalkowski, “HI-TIQA: an interactive question answering system: A preliminary report,” in *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, C.-Y. L. Abraham Ittycheriah, Tsuneaki Kato and Y. Sasaki, Eds., 2003.
- [6] E. M. Voorhees, “Overview of TREC 2003,” in *TREC*, 2003.
- [7] F. Andén, L. Degerstedt, A. Flycht-Eriksson, A. Jonsson, M. Merkel, and S. Norberg, “Experiences from combining dialogue system development with information access techniques,” in *New Directions in Question Answering*. AAAI/MIT Press, 2004.
- [8] M. De Boni and S. Manandhar, “Implementing clarification dialogues in open domain question answering,” *Journal of Natural Language Engineering*, 2004, accepted for publication.
- [9] K. Inui, A. Yamashita, and Y. Matsumoto, “Dialogue management for language-based information seeking,” in *Proc. First International Workshop on Language Understanding and Agents for Real World Interaction*, 2003, pp. 32–38.
- [10] E. M. Voorhees, “Overview of TREC 2001,” in *TREC*, 2001.
- [11] T. Kato, J. Fukumoto, and F. Masui, “Question answering challenge for information access dialogue – overview of NTCIR4 QAC2 subtask 3,” in *Working notes of the Fourth NTCIR Workshop Meeting*, 2004.
- [12] C. Hori, T. Hori, H. Isozaki, E. Maeda, S. Katagiri, and S. Furui, “Deriving disambiguous queries in a spoken interactive ODQA system,” in *Proc. ICASSP*, vol. 1, 2003, pp. 624–627.
- [13] H. Bunt, “Dialogue pragmatics and context specification,” in *Abduction, Belief and Context in Dialogue*, H. Bunt and W. Black, Eds. Benjamins, 2000, pp. 81–150.
- [14] E. P. Salonen, M. Hartikainen, M. Turunen, J. Hakulinen, and J. A. Funk, “Flexible dialogue management using distributed and dynamic dialogue control,” in *Proc. of ICSLP 2004*, 2004.