# Experiences from combining dialogue system development with information extraction techniques

Arne Jönsson, Frida Andén, Lars Degerstedt, Annika Flycht-Eriksson, Magnus Merkel, and Sara Norberg
Department of Computer and Information Science
Linköping University, Sweden

**Abstract**

Traditional Q&A systems are efficient at interpretation of single questions and extraction of the corresponding answer from unstructured texts, but cannot handle many of the interaction features that make dialogue systems so efficient for information access. Dialogue systems, on the other hand, can handle connected dialogues, but are normally developed to access structured data, often stored in databases.

The challenge is therefore to combine these areas of language technology research and develop dialogue systems that can access information from unstructured text documents. A first step towards this goal is to use information extraction techniques that pull out relevant information from textual documents to compile unstructured information to a database. This might sound as a straightforward endeavour, but in practice, it involves a number of research issues, such as, handling different ontological perspectives, dealing with information gaps, inference both inside the dialogue and in the interpretation of source documents, etc.

We have addressed this combined research issue of utilising information extraction techniques to automatically create structured databases from unstructured documents to be accessed by dialogue systems. A system, BirdQuest, has been developed based on a bird encyclopaedia from which information was extracted and transformed to a relational database. In the paper we present the system architecture, its components, and evaluations from the perspectives of users of the system and the development of a dialogue system that access a database created automatically utilising information extraction.

## 1 Introduction

In the field of Question Answering, Information Extraction (IE) techniques have been used successfully when it comes to retrieving answers to fact-based questions [28, 23] from unstructured information, but the Q&A approach has yet not reached the level of sophistication for handling connected dialogue as is present in dialogue systems tailored to background systems with structured data. Dialogue capabilities allow, for example, for more precise formulation of information requests and follow-up questions. A desired improvement to future Q&A systems would therefore be to incorporate advanced dialogue management in order to facilitate more natural interaction between the system and its users [7].

Dialogue systems, e.g. [24, 9, 12, 3, 5] are designed for interaction with background information in the form of ready-made structural information, mostly relational databases. Most of the information available in electronic formats is, however, not found in databases; the vast majority of information comes as text, making up huge sets of unstructured information in natural language. This means that dialogue systems either have to be adapted to handling unstructured information on the fly, as in Q&A systems, or be provided by structured information in database formats. The latter could be performed with information extraction techniques that pull out relevant information from textual documents and compile it to a usable database.

In this article we investigate how the pre-compiled approach works, by developing a dialogue system where information is automatically extracted from a text book which then can be accessed


Figure 1: The graphical interface of BirdQuest.

through natural language dialogue. This allows us to (a) use and evaluate existing methods for information extraction and dialogue, and (b) to investigate how much of the development must become shared for both enterprises as a "minimal assessment". We present one such system, BirdQuest, where sharing of knowledge resources is utilised. It was implemented by an incremental approach for ontology creation, information extraction and interactive dialogue following an iterative scheme. This can also be seen as a first step towards the more advanced goal of dialogue with dynamically extracted text information.

The article begins by presenting BirdQuest and the overall architecture, Section 2. Section 3 presents the design and construction of the ontology. In Sections 4 and 5 we present the two components, information processing and dialogue management respectively. We have conducted a small evaluation of BirdQuest which is presented in Section 6. Finally, we discuss implications for system development in Section 7.

## 2 BirdQuest

BirdQuest was developed for a web site where people, watching nature programs on TV, can ask questions related to the TV program, in this case questions on Nordic birds. A screen shot of BirdQuest is seen in Figure 1.

The BirdQuest application is built using the dialogue system framework malin, a code and application development framework that has evolved for a long time over several earlier dialogue system projects. The code base of malin is adapted partly by additions of new content for its information resources, such as a grammar, and partly by adjustments and pattern re-use of source code (currently Java). Application development follows a similar iterative basic scheme for any application but may vary depending on certain details depending on the domain. In particular, we

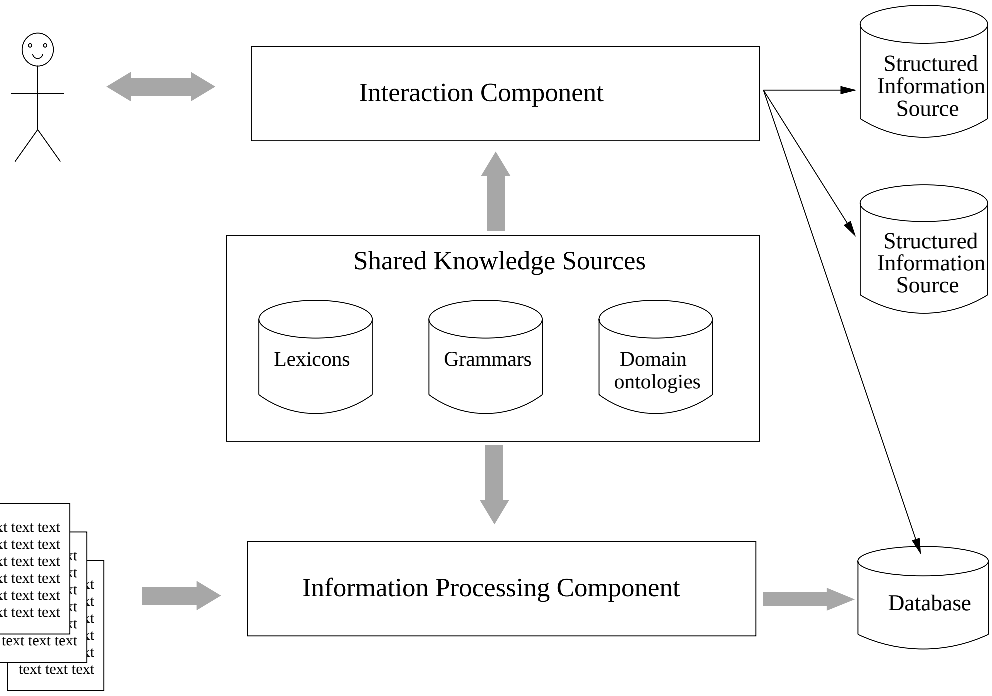


Figure 2: An architecture for systems that combine information extraction with dialogue interaction. The figure shows the different components and the shared knowledge sources used by the components.

have used the MALIN framework together with Wizard of Oz WOZ tests [10] and mixed initiative dialogue domains.

For BirdQuest the MALIN framework is set up to function as the basis of an information processing component together with an interaction component, as shown in Figure 2. The information processing component takes collections of unstructured or semi-structured documents and transforms them into structured information that can be used by the interaction component during interaction with the user. The unstructured information on birds in a bird encyclopaedia [26] was marked up as XML entities using simple patterns for tagging named entities, such as bird names, colours, measurements, etc. This information was then used as a basis for information extraction to create and fill a database with structured information. The transformation is based on information extraction techniques, and the documents are analysed in several stages going through lexical and morphological, syntactical, and domain analysis, in each step adding more structure to the documents. The interaction component is responsible for the dialogue interaction with the user. It collaborates with the user to produce a question and accesses the structured information sources to retrieve an answer to that question.

As a basis for the processing of documents and user questions a number of knowledge sources are utilised. Some are highly specialised and only used by one or a few sub modules of a component, for example the dialogue model in the interaction component, while others are more general and used for several tasks in both components e.g. the domain ontology.

Shared data sources are used for communication between the two components. A common agreement on lexical, grammatical and ontological knowledge is therefore required. The important starting point for the combined information extraction/dialogue system development of BirdQuest was a dialogue corpus collected for this domain.

Adding dialogue capabilities, such as clarifications and connected dialogue, facilitates formulations of complex and connected requests. The dialogue component utilised in BirdQuest adds dialogue history, contextually sensitive pragmatic interpretation, and clarification initiatives to the standard executional phases of Q&A.

Figure 3 shows the dialogue in Figure 1 translated to English. U1-S2 shows a complex search in the database. In utterance U3 the user asks for the property "size" which, as seen in S4, has two different aspects, "wingspan" and "length". In U5 the user uses a pronoun to refer to the object in focus, i.e. common kestrel, and in U7 the user implicitly refers to the same property, i.e. "eggs", but for a different bird, "loon". However, there are many loons so BirdQuest asks the user to select one in S8. The user does that in U9 and the user gets a response based on the property provided in utterance U5 and the bird in U7-U9.

| | |
|---|---|
| U1: | Which birds nest in Africa |
| S2: | Montagu´s harrier, common kestrel, Eurasian hobby, Kentish plover, Gull-billed tern, Stock dove and European Bee-eater |
| U3: | How large is the common kestrel |
| S4: | - A common kestrel is about 35 cm long from beak to tip of the tail.<br>- A common kestrels wingspan is about 80 cm. |
| U5: | How many eggs does it lay |
| S6: | A common kestrel lays approx. 3-6 eggs |
| U7: | Loon |
| S8: | Select one of the following loons: Yellow-billed loon, black-throated loon, red-throated loon and great northern loon |
| U9: | red-throated loon |
| S10: | A red-throated loon lays approx. 1-2 eggs |
| U11: | Where does it nest |
| S12: | Red-throated loons geographic nesting area: Is the one of the four loons that nest northernmost |

Figure 3: English translation of the Swedish dialogue in Figure 1.

The architecture makes the interface between the components minimal and black-box, which means that existing MALIN resources and the information extraction tools can be kept fairly intact. Adjustments of the source code for BirdQuest mostly comes from increased complexity in ontological information. Application development becomes more complex mostly due to more elaborate retrieval of domain data, and how to foresee what should be extracted to facilitate robust and relevant dialogue coverage.

In what follows the components in Figure 2 will be further described.

# 3 Domain ontology

The ontology is the most complex knowledge source, and it is shared by both components. The term ontology is used very differently in various areas of computer science, ranging from simple taxonomies, meta data schemes, to logical theories. A general and commonly used definition, adopted here, is that *"An ontology is a formal, explicit specification of a shared conceptualisation"* [14]. A more practical view is to consider an ontology as *"a world model used as a computational resource for solving a particular set of problems"* [22].

The most basic type of concept to include in ontologies is *objects*. Many ontologies have thing or entity as a top node. Almost as common are *event* and *process*. In some cases events and processes are seen as a subclass of entities but often they are distinguished from objects, (cf. Sowa's ontology [25] and Mikrokosmos [21]).

To define and describe these two types of concepts, *properties* are used. They are often implicit, for example, seen as part of a thing's internal structure and represented by a slot in a frame. The frame type of representation has been used in dialogue systems to represent an object and its properties, for example, a trip with a departure and destination location and departure or arrival time (cf. [17, 5]). In traditional information extraction, frames have been used in a similar way to represent events and the roles of the objects related to these (cf. [8]).

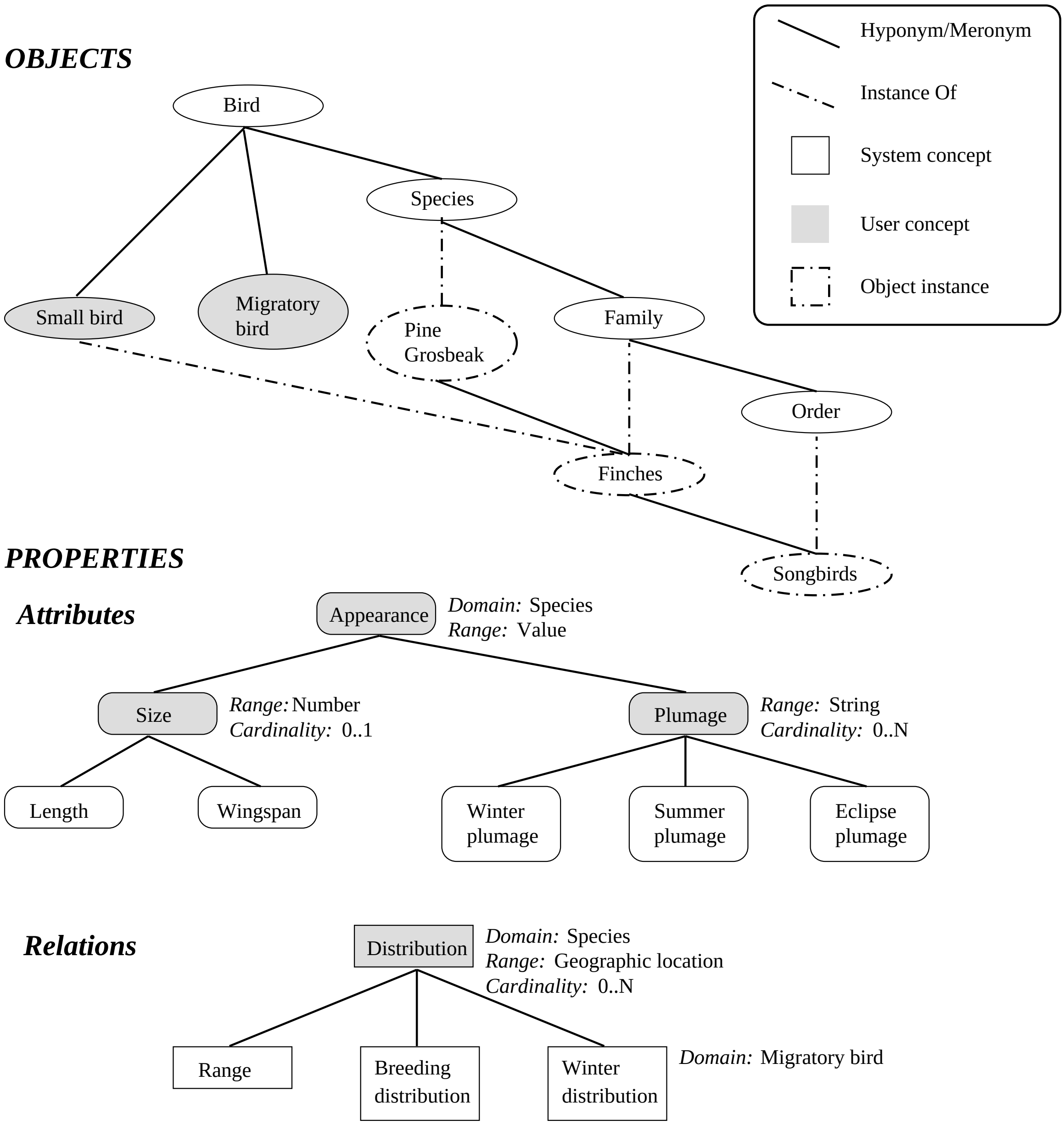


Figure 4: A part of the integrated ontology representing the conceptualisations of both bird encyclopaedia and users.

Although explicit representation of properties is not common, they do exist. For example, in Mikrokosmos [22], properties, which include attributes and relations, are represented both as explicit concepts and slots in objects and events. Explicit representation of properties can be motivated from a dialogue management perspective, since it facilitates contextual interpretation of requests and aid clarifications. In the MALIN framework, objects and properties are used for modelling of focal information, implemented by a slot-and-filler structure.

The primary purpose of the ontology in BirdQuest is to support the interaction component's tasks of cooperatively formulating information requests together with the user and of accessing and retrieving the requested information. It should also model the type of information to be extracted from the source document and stored in the database. Since the bird encyclopaedia almost exclusively contained information of a factual character with focus on objects, the BirdQuest ontology was restricted to modelling this type of information in terms of the objects and their properties, which were further divided in attributes and relations, leaving out events/processes.

Users often have different perspectives of the domain in question compared to the concepts expressed and used in reference material written by domain experts, such as a bird encyclopaedia. For instance, in the case of a bird anatomy, an expert would have specialised terms to refer to different kinds of feathers whereas the novice would use more every-day descriptions. Thus, we need to identify the different views on the ontology. For BirdQuest a user conceptualisation was developed from a question corpus and a system conceptualisation was based on the textual documents. The former captures the concept utilised for dialogue interaction and the latter concepts needed to translate user requests to SQL requests.

The organisation and structure of the bird encyclopaedia were taken as a starting point for identification of system-oriented ontology concepts. It used the K H Voous system for dividing birds into orders, families and species. For each of the categories information about certain properties was presented, most of which were species specific. The book was manually analysed to identify the objects, attributes, and relations relevant for the purpose of information extraction. The analysis gave a total of 8 objects (various groupings of birds, and geographical locations), 30 attributes and 3 relations.

The corpus used to detect user-oriented concepts consists of 264 questions about birds. It was collected by The Swedish public service television company on a web site for one of their nature programs, where the public could send in questions. The analysis of the corpus focused on questions that were deemed as within the boundaries of the application leaving out, for example, questions concerning veterinarian treatment of birds or explanations of behaviours. The analysis of the remaining questions in the corpus revealed that the users' view of the domain in most cases corresponds to the one found in the encyclopedia, but a small number of new categories and several new attributes were identified.

Taking the system-oriented conceptualisation as a starting point the new categories of birds found in the question corpora were added. Allowing multiple inheritance, new links between existing categories and new categories were included. Note, for example, how the new category "Small bird" is introduced and a new link is added to "Finches" in Figure 4. Conflicting concepts were resolved in various ways. One example is the use of the word "large" in the utterance, *Which is the largest bird?*. There is not a single corresponding concept for the word "large" in the domain ontology, instead a user ontology concept, "size", is introduced which in turn is mapped to the two domain ontology concepts: "wingspan" and "length", cf. S4 in Figure 3. This mapping is domain dependent and consequently not done in the lexicon but in the ontology, as being large does not imply, for instance, having large wingspan in all domains. A problem encountered was the user-introduced categories for birds: "Small birds", "Migratory birds", "Sedentary birds" and "Birds of prey". The bird encyclopedia contained no or sometimes ambiguous information concerning the first three of these. There is no clear definition of a "Small bird" and for many species the migratory behaviour varies for young and old birds. Thus only "Birds of prey" had instances in the implementation. With 4 new objects and 6 new attributes introduced by the analysis of the question corpus, the resulting ontology contained 12 objects, 36 attributes and 3 relations.

The ontological knowledge was represented in two separate parts, one containing the concepts and their taxonomical relations, and one holding the facts, i.e. object instances and their taxonomical relations. The factual part was generated semi-automatically by extraction of instances from the bird encyclopedia.

Each concept had a unique name, a definition in natural language, and a tag stating if it was system or user derived, i.e. if it came from the bird encyclopedia conceptualisation or the question corpus. Attributes and relations also had domain and range restrictions, which stated what type of objects they were applicable to. This is illustrated in Figure 4 where the relation "Winter distribution" has domain "Migratory bird" and range "Geographical location". There were also cardinality restrictions for the attributes and relations.

# 4 Information extraction

The information processing component was constructed in order to extract the relevant information from the unstructured text. The design of this module was based on the analysis of the question corpora as well as an analysis of the text.

From the analysis of the text it became obvious that there were several difficulties in handling the information in the bird encyclopaedia. The encyclopaedia contained varying amounts of facts about the birds; some of the species were described in detail while descriptions of other species had very little information. Also, birds were described by their most characteristic features, which led to omission of many traits. A related problem was that the text was complemented by pictures of the birds and maps of their distribution. Because of this, text often referred to maps and images, and left out the information that could be seen there, e.g. *Outside the map the Black-throated diver nests in Russia and China.*

The goal of the information processing component was to identify and extract the objects and properties in the text that were relevant to the user questions. This included identifying species and families, and finding different types of properties on the family level as well as on the species level. It also included finding relations in the text, like which family a species belongs to. The selection of what information to extract was guided by the collected question corpus. The objective was to only fill the database with relevant information and ignore text segments that did not meet the needs of the users. The information processing component extracts atomic values as well as segments of text, with the purpose that text segments could be provided as answers to the user. Atomic values were extracted only when the question corpus indicated that comparisons between birds would be needed. The possibility of extracting atomic values exists for all the records in the database.

The Information Extraction used for BirdQuest is a process in six steps, see Figure 5. The unstructured information text is given as input to the information processing component, see Figure 6, along with a specification of what information to extract.

The first part of the IE process is tokenization and pre-processing of the text. In this module the text is processed to facilitate the parsing. This means that some of the characters in the text are replaced with others, for example the FDG parser [27] had trouble parsing sentences with semicolons correctly, and these characters are therefore replaced with colons.

The next phase in the processing is the syntactic parsing of the text where functional dependencies and part-of-speech tags for the words are obtained. The results of the FDG parse is transformed to XML format, using an XML conversion package [1]. This is done to enumerate the sentences and facilitate text searches. All the information from the FDG-parse is preserved in the XML format.

Named Entity Recognition is the next step of the IE processing. In this particular domain focus lies on finding the names of the species, families etc. The Named Entity Recogniser utilises a lexicon to identify and tag geographic sites, colours, body parts, names of species, and families, Latin names, food, etc, see Figure 7. The lexicon was automatically created in an earlier step through the use of Perl scripts that utilised the orthographic environment and syntactic information associated with each word in the text to find the words of interest. The lexicon was inspected manually to ensure that it was correct, and the synonyms that were discovered in the analysis of the text were added.

After NE recognition the Segment Retrieval module finds and tags the parts of the text relevant to each species. This is done using the tags from the previous module and through analysis of the orthographic environment of each paragraph.

In the Pattern Matching phase three types of patterns are used to find and extract the facts in the text: patterns on surface level, on syntactic level and on a semantic level. The surface patterns handle simple patterns on the word form level that are recurring in the text, for example "Maximum age: 32 years". The syntactic patterns are used to find facts trough their syntactic features, for example, to find the colour or shapes of certain body parts, through the use of functional dependencies and part-of-speech properties in the sentences. The semantic patterns are patterns like "MOVES TO GEOGRAPHIC_SITE TIME_OF_YEAR". In a number of cases these

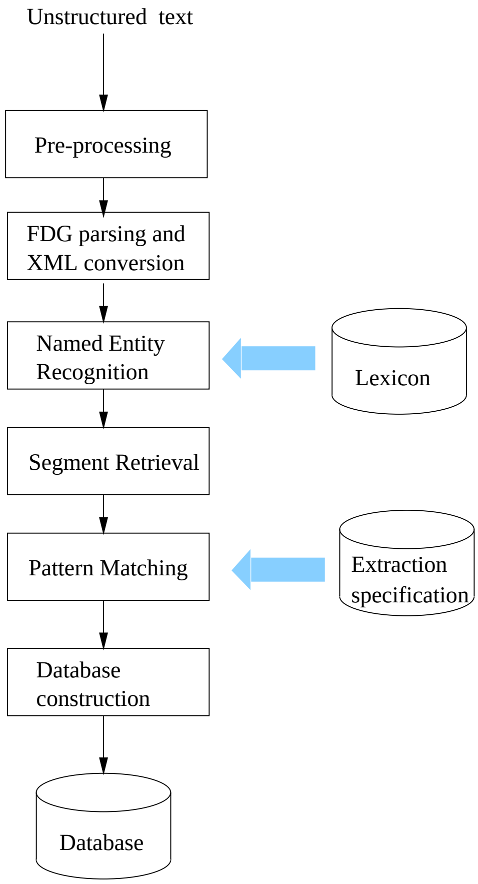


Figure 5: The information processing component. The figure shows the six information extraction steps.

```
Black-throated diver
Gavia arctica
58-73 cm, wingspan 110-130 cm.
In breeding plumage the head is gray and the throat is black,
the sides of the throat striped in black and white.
```

Figure 6: Short extract of the beginning of a text passage describing a particular bird in the bird encyclopaedia “Nordens fåglar” (translated from Swedish).

```
<NAME> Black-throated diver </NAME>
<LATIN> Gavia Arctica </LATIN>
58-73 cm, wingspan 110-130 cm.
In breeding plumage the <BP>head </BP > is <COLOUR > gray </COLOUR>and
the <BP>throat</BP> is <COLOUR>black</COLOUR>, the sides of the <BP>
throat </BP> striped in <COLOUR>black</COLOUR> and
<COLOUR>white</COLOUR>.
```

Figure 7: Result, simplified, of applying named entity extraction to the bird encyclopaedia.

different types of patterns are combined. Patterns are compiled into one set of pattern instructions that is run in batch. The properties that can be extracted from the text are all the properties that are leaf nodes in the ontology. Figure 8 shows the text in Figure 6 after these stages.

```
<NAME> Black-throated diver </NAME>
<LATIN> Gavia Arctica </LATIN>1
<MIN_LENGTH>58</MIN_LENGTH>-<MAX_LENGTH>73</MAX_LENGTH>
<MIN_WING>110</MIN_WING>-<MAX_WING>130</MAX_WING>
<BREEDING PLUMAGE>In breeding plumage the <BP>head </BP > is <COLOUR >
<COLOUR_HEAD>gray</COLOUR_HEAD> </COLOUR>and the <BP><throat</BP> is
<COLOUR><COLOUR_THROAT>black<COLOUR_THROAT></COLOUR>, the sides of the
<BP> throat </BP> striped in <COLOUR>< COLOUR_THROAT> black</
COLOUR_THROAT> </COLOUR> and <COLOUR> <COLOUR_THROAT> white</
COLOUR_THROAT ></COLOUR>.</BREEDING PLUMAGE>
```

Figure 8: Result, simplified, after stage four and five of the IE process, Segment Retrieval and Pattern Matching, yielding a more fine grained specification of the properties given in the text.

After the pattern matching has taken place the results are compiled to a text file. This is then transformed to relational database format to serve as background information in the dialogue system, see Figure 9. The relational database allows BirdQuest to handle more complex requests concerning relations and comparisons, such as *Which is the largest bird* or *Are there any birds that lay more eggs than this one?*. The database contains two main tables, one for bird species and one for bird families.

```
NAME: Black-throated diver
LATIN_NAME: Gavia arctica
MAX_WING: 130
MIN_WING. 110
MAX_LENGTH: 73
MIN_LENGTH:58
BR_PLUMAGE: "In breeding plumage the head is gray and he throat is
black, the sides of the throat striped in black and white"
```

Figure 9: Excerpt from the database, showing both atomic values and text strings.

As can be seen in Figure 9, some of the extracted features shown in Figure 8 are not present in the final database. This is because the high degree of variation among these facts makes them difficult to be of any use to the dialogue system. To reason about the colours and shapes of body parts, for example, the ontology would have to be extended to include a representation of what body parts are meronyms to others. There are over 100 different body parts to choose from. Furthermore, for each species only the most characteristic body parts are described, seldom more than three parts, and the variation is huge among these. Colours are also often described in relation to other birds, e.g. *The Blyth's Reed Warbler looks like the European Reed Warbler and the Marsh Warbler but the upper parts' brown colour is slightly greyer.*, and an inference tool would then need to assemble information from these different species in order to provide a complete colour description.

The text includes many sentences that are not grammatically complete, for example many sentences lack subjects, which caused difficulties for the FDG parser. The syntactic errors stemming from this led to difficulties in the extraction, since it relies heavily on the FDG parse being accurate.

# 5 Dialogue and domain knowledge management

The Interaction component processes utterances in phases, see Figure 10. During the pragmatic interpretation phase, dialogue history and local focus are used to create a complete dialogue move from the incoming linguistically analysed user utterance. This move is subsequently handled, during the task handling phase, either by generating a clarification initiative or by accessing and retrieving information from the BirdQuest information sources (the database or the domain ontology), depending on how the dialogue move is classified. Task handling results in a system utterance structure, which is on a format compatible with that of the user dialogue move, in order to facilitate later focus inheritance. During the generation phase, the system utterance is transformed into the multimedia surface format, which yields appropriate formatted textual results, images and sounds for the user interface. In the memorisation phase, current user and system moves are recorded in dialogue memory.

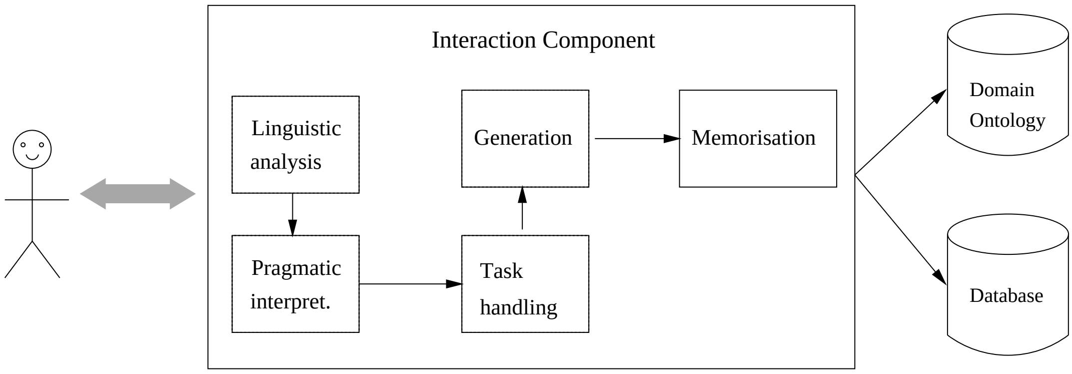


Figure 10: Interaction component based on the MALIN framework. The figure illustrates the various phases gone through to process an information request from the user.

The dialogue model used in MALIN classifies the discourse segments by general dialogue act categories, the act type, such as *question* (Q) and *answer* (A), rather than specialised (cf. [16]), or domain related [2]. Dialogue management instead utilise the focal parameters to control interaction (cf. [18, 12, 19]). The dialogue history is represented by user and system dialogue moves with a parameter termed objects, which identify a set of primary referents, and the parameter properties which denote a complex predicate ascribed to this set.

By analysing the question corpus of BirdQuest and the information text, the names and value domains of the objects and properties parameters where selected, based on the domain ontology. In reality there are many more potential objects and properties both in the information text and in the question corpus. The selection of objects was in some ways straightforward since the obvious objects are birds. Other objects, like the kinds of food that is eaten by the birds, (e.g. fish), or places the birds stay at, (e.g. countries they migrate to during wintertime), were not chosen mainly because there was very little or no information about these potential objects in the information text so there would be no information about them for the system to present.

There is, however, still a point to handling the above conceptual entities of the system. Since the information is put into a relational database it is possible to make inferences about the data. Fields and records in the database can be searched and matched against a text string by using regular expressions (and/or by simple annotations of text). Texts about the birds migration places and foods contain many objects. By including some of these and interpreting them as values they can be matched against the text fields in the database. This makes it possible to "stretch" the scope of the extracted text segments, for instance, to derive the names of the birds that migrate to a specific place or eat a specific type of food from the food text field in the database, as in the dialogue in Figure 11.

Managing the domain objects is still quite problematic, since there is at present no ontological information available for values, which means that the system is not capable of making inferences about the values. In the example above search is only made for the concept "small rodents", ignoring instances of small rodents, for example mice.

Using information extraction techniques means that knowledge sources can incorporate more

U1: Which birds eat small rodents?

S1: Kite, western marsh harrier, hen harrier, Montagu's harrier, Eurasian sparrow hawk, common buzzard, red-footed falcon, parasitic jaeger, long-tailed jaeger, barn owl, Eurasian pygmy owl, red-backed shrike, Eurasian jay, and magpie.

Figure 11: Dialogue illustrating how text fields in the database are used to derive implicit information.

elaborate linguistic content, such as ontological data. Consequently all dialogue capabilities of BirdQuest must be adjusted to handle and exploit such resources in its various phases and steps of execution. For pragmatic interpretation, this amounts to handling properties of various levels of ontological abstraction. For domain knowledge source management it amounts to handling both information made explicit in the data model, and derived results. For user dialogue acts that are still under-specified after contextual data has been added, ontological information and data models help to give valuable request for more information and point out what the system can handle.

## 5.1 Pragmatic interpretation

Pragmatic interpretation uses a dialogue memory that holds the dialogue history. It is implemented by a stack of user/system dialog moves which uses so-called unwind-pointers to keep track of where clarification dialogues start. The pragmatic interpretation process consists of the following sub-filters:

- **instantiator**: instantiates anaphoric references with actual values.
- **normaliser**: modifies the dialog act parameters asserted at linguistic level, and fills in default values, if possible in the given context.
- **integrator**: if the act type is an "answer" to a clarification request, then the answer is integrated to form a new request using the clarified incomplete request.
- **focuser**: completes the objects and properties parameters of current user , using previous focus of the dialogue history.

The instantiator uses linguistic data included in an object parameter description. The integrator "fills in" the missing pieces of data of the clarified request, based on a structural analysis of the incomplete request. The focuser inherits names and values from the objects and properties parameters only when no new entities of these parameters are given by the user. Otherwise, the set of entities in focus is considered as "new", w.r.t. that parameter. A distinction is also made between user requests with or without relative or extreme operators, such as "more" and "most". In the latter case a particular focus operation is performed to derive the latest "set of" objects in focus, rather than simply the previous objects themselves.

Each time a new domain request has been successfully created by the focuser, the dialogue memory is unwinded to a previous unwind-point, if such exists. This means that the user can override a system clarification request by giving a new complete request. An exception to this rule is for user probing questions of type "which/isa" and "members of", which are allowed also within a clarification sub-dialogue, and therefore should not unwind the dialogue context. For example, consider how the properties in focus stay constant for the sub-dialogue of Figure 12.

## 5.2 Domain knowledge lookup

Before the pragmatically interpreted dialogue move is executed, it is further pre-processed during the task handling phase. Properties of the move can be too vague to be directly executed, such as a property "size" derived from the sentence *How large is the magpie?*. The property hierarchy of the ontology is used to transform such user-related properties into more concrete system-related

U1: What do birds of prey eat
S2: Select one of the following birds of prey: Hawk-like birds of prey, ospreys and falcons.
U3: What falcons are there
S4: The following species belong to the family falcons: common kestrel, merlin, peregrine falcon, Eurasian hobby, gyr falcon, and red-footed falcon.
U5: Take ospreys
S6: Osprey: Eats only fish caught with the claws after a violent crash dive in the water. Catches often medium size fish important fish are pike, perch and roach.

Figure 12: Dialogue that illustrates how context is kept the same during system-initiative sub-dialogues. The preferred property in focus of U5 is "food", which comes from utterance U1. That is, the user probing question in U3 that introduces an "isa" property does not change the focus of the sub-dialogue. Without the ongoing sub-dialogue, U5 would have inherited its property from U3. U5 would then have been interpreted as the request "what ospreys are there", instead of "what do ospreys eat".

specialisations of the request. For instance, the "size" is transformed downwards to find the children "wingspan" and "length", see U3-S4 in Figure 3.

The reason for doing this transformation from sentence to system-related properties in two steps—thus making the user-related properties explicit in the system—is to be able to store and use these properties for future focus inheritance, which then can be instantiated differently depending on the current context. The size of an egg is perhaps not the same as the size of a bird in terms of system-related properties, but the property "size" can be inherited anyway.

A fully specified dialogue move from the user is processed using the two sources of information of BirdQuest: the ontology and the database with the extracted texts. The data lookup is performed using a collection of SQL templates that are combined to handled different classes of requests: request for object attributes of a given object, request for any object that have certain property values, and request for objects which are extreme in some way. The templates for retrieval of objects normally iterates over the properties in the properties parameter and merge the database results, which makes the setup general (see Figure 13 for an example).

After a user dialogue move has been executed the results are merged into a responding system dialogue move. The format of the system move is also based on the MALIN format and the focal parameters. The created move is post-examined, and if the size is either too large or empty the move is transformed into a correcting system message instead.

U1: Which is the largest swan
S2: Whooper swan has largest wingspan. Mute swan and whooper swan are longest.

Figure 13: Iterative retrieval of a collection of objects with maximal property values.

### 5.3 Clarification initiatives

During the task handling phase, both the ontology and the database is used to help clarify under-specified requests. This can be seen as a pre-processing step of the task handling phase that does not yield a data lookup for the request in the normal way. The information source lookup is here integrated with the pre-processing, which is different from the case of the complete requests. The clarification lookups have their own methods for traversing the ontological taxonomies and their own SQL templates.

A user dialogue move is considered complete only if it contains at least one object and one property. If the move only contains either one object or one property the ontology is used to collect information to create a clarification message to the user.

For instance, if the request only contains one object, and no property, the ontology is used to collect all relevant properties from the property hierarchies. The suggested ontological concretisations are checked for values in the database, since the database may lack certain values for particular objects. The alternatives that do have values are presented to the user as suggestions, cf. Figure 14.

U1: bullfinch
U2: You can have information about a bullfinchs´ eclipse plumage, flight, food, young, Latin name, plumage, greatest age, appearance, breeding, breeding place, migration habits, call and distribution. What do you want information about?

Figure 14: Dialogue with BirdQuest illustrating clarification request for species.

If the request only contains a property but no object, the user is asked to specify a bird within ontological domain of the property, as seen in Figure 15.

U1: food
U2: Please select a bird family.

Figure 15: Dialogue with BirdQuest illustrating clarification requests for bird families.

If a request contains both an object and a property the ontology is used to check whether the category which the object belongs to, i.e. family, species etc., is the correct domain for the property. The domain is set according to which level the information is in the database. If it is not, objects that are children to the given object in the object hierarchy that correspond to the domain are collected by traversing the object hierarchy. Again, only entities that really have database values are selected, using a database lookup for existence of values. Objects in the category of the domain of the property are collected and presented to the user as suggestions for a specification of the request, as exemplified in Figure 16.

This method is also applied if the database lacks information about the given property for the given object, provided that the object is not a leaf node, in the object hierarchy of the ontology.

U1 What do corvids eat?
S2 Select one of the following corvids: magpie, rook, Eurasian jay, spotted nutcracker, Siberian jay, carrion crow, common raven, and western jackdaw.

Figure 16: Dialogue with BirdQuest illustrating search in the ontology.

## 6 User evaluation of BirdQuest

BirdQuest is intended to be used by casual users without previous experience of dialogue systems or extensive knowledge of birds. The system was therefore evaluated in a walk-up and use situation similar to a real use situation during a day when the public was invited to the university. In that respect the situation resembles that of Gustafson & Bell [15], though slightly more controlled.

We had six machines running BirdQuest during 2 hours and 30 minutes and collected dialogues from 27 users. They received minimal instructions in advance, they were only told that the system can answer questions on Nordic birds, that it understands Swedish, and that the dialogue would be recorded.

The resulting corpus consisting of 27 dialogues have a total number of 518 user utterances, with a mean of 19 for each user. However, with individual differences, for instance, three users posing more than 40 utterances to the system and three users posing less than 5.

Table 1: User utterances

| | No of utterances | Percentage of user utterances |
|---|---|---|
| Interpretable Requests | 189 | 37% |
| Cooperative CR Responses | 55 | 11% |
| Uncooperative CR responses | 11 | 2% |
| Out of scope | 121 | 23% |
| Mis-interpreted | 141 | 27% |

Table 2: System utterances

| | No of utterances | Percentage of system utterances |
|---|---|---|
| Successful resp. | 180 | 35% |
| Clarification req. | 70 | 13% |
| Incorrect resp. | 15 | 3% |
| Incorrect focus | 16 | 3% |
| Error message | 240 | 46% |

## 6.1 Corpus annotation and initial analysis

As we had no predefined tasks we did not have a situation that allowed for a controlled evaluation, as e.g. PARADISE [29] or PROMISE [6]. Instead we used a combination of quantitative and qualitative approaches to analyse the collected dialogue corpus. The dialogues were tagged in order to provide statistics over successful and problematic information exchanges [13].

The user utterances were categorised as in Table 1 and the categorisation of responses from BirdQuest is presented in Table 2.

Table 1 shows that approximately half of the users utterances (48%) were involved in successful information exchanges where the user initiated an information request or answered a clarification request from the system. We also see that 25% of the user´s utterances are erroneous in some way and that BirdQuest failed on 27% of the utterances.

From Table 2 we see that BirdQuest presented 180 successful responses. A successful response is a response where BirdQuest presents information found in the database. A response where the bird encyclopaedia does not include the information and BirdQuest responds e.g. *Information on wingspan is missing for magpie.* is, however, also considered successful. The reason being that BirdQuest successfully accessed the database and presented whatever information was there, including cases where there was no information in the database. Among the 180 there are 55 such responses, so they are not rare, and shows one of the many interesting problems we encountered in the development of a dialogue system based on information extraction from a text book. We will further discuss BirdQuests performance below.

### 6.1.1 Dialogue features

Notable is that a fair amount of the dialogue moves involve clarifications. The system initiates 70 clarification sub-dialogues in order to transform a vague information request to a specific question, as exemplified by the excerpt in Figure 17[1].

The basis for this type of clarification is domain knowledge collected from the domain ontology. Utterance U17 is under specified since the object, the bird family "Waterfowl", can refer to a number of different species of birds, and the property "Appearance", is vague. To pose clarification question S18, information about which species belong to the given family is gathered from the

[1] All examples in this section are translations of excerpts from the Swedish dialogue corpus, reflected by utterance numbering.

U17: What do waterfowl look like?
S18: Choose one of the following waterfowl: barnacle goose, tufted duck, bean goose, whooper swan, gadwall,...
U19: gadwall
S20: You can receive information about the size and plumage of a gadwall. Please specify what you are interested in.
U21: size
S22: - A gadwall is about 56 cm long from beak to tip of the tail.
- Information about wingspan is missing for gadwall

Figure 17: Successful clarification sub-dialogues.

ontology and the user is asked to chose one of them. Next, in S20, the ontology is accessed to retrieve the sub-properties of appearance. When the user has chosen a specific property (U21) the request is sufficiently specified. The ontology is used to find the sub-properties of "Size" and these are then used to access the database and the result is presented to the user (S22).

The users responded cooperatively to 55 clarification requests from the system and incorrectly 11 times. A typical example of the latter is seen in Figure 18.

S22: You can receive information about size and plumage of a Blue Tit. Please specify what you are interested in.
U23: blue tit

Figure 18: Uncooperative response to clarification request.

Dialogue management, such as clarification sub-dialogues, thus plays an important role for the performance of BirdQuest.

Contextual interpretation and dialogue history management are other important dialogue phenomena from malin that are frequently utilised in the dialogues. Managing dialogue history is, however, not trivial. There are 16 cases in the corpus, termed Incorrect focus in Table 2, when BirdQuest presents doubtful responses because of how dialogue history is handled.

Normally, sequences of utterances are contextually interpreted utilising the dialogue history and a successful response can be presented to the user, cf. Figures 3 and 12. Consider, however, the dialogue fragment in Figure 19[2]. Here, the user is trying to shift focus and thus the property "feed" should not have been inherited from the previous question (U11). A more appropriate response in S18 would have been *You can receive information about a song thrush's flight, feed, breeding location, appearance,... Please specify what you are interested in.*

U11: What do the honey buzzard eat
S12: Information about feed for a honey buzzard:
Main feed consists of honeycomb,[...]
U13: What birds can imitate speech
S14: Failed to interpret the question. Please try again
U15: Talking birds
S16: Failed to interpret the question. Please try again
U17: Song thrush
S18: Information about feed for a song thrush:
Worms, larvae and mollusc etc[...]

Figure 19: Problematic dialogue history.

The problem of dialogue history and focus shifts does not have a simple solution. Many solutions are based on identifying relations between concepts [20, 18, 12]. However, as we assume that it

[2]The Swedish name of Song thrush, (Sw. taltrast), can be literally translated as "Talking thrush"

is a good idea to keep items in focus even after user utterances with no previous relation to the current focus [19], such mechanisms are not applicable here.

#### 6.1.2 Utterances out of scope for BirdQuest

Approximately half of the non-successful user utterances (23% of all user utterances) were questions that BirdQuest will never be able to answer, as the users "fall out of the role or purposely misuse the system." [6].

Out of Scope utterances include user requests for information that is outside the scope of the application, such as *How do you kill crows?*, or socialisation utterances [15] such as *How are you?*. Utterances can also be out of the database' scope, e.g. *How high does a magpie fly?* is such an utterance since there is no information on how high birds fly in the bird encyclopaedia.

Users often posed questions that were outside the database coverage. A successful interpretation of such a request and an appropriate response is seen in the dialogue in Figure 20:

U1: What is the wingspan of an Albatross?
S2: There is no information on non-Nordic birds. The available birds are listed in the menu "Birds a-ö". Please try again

Figure 20: Request out of scope for BirdQuest.

The same strategy could be used to handle similar questions concerning non-Nordic birds like canaries and ostriches. However, that would require addition of information of non-Nordic birds to the ontology collected from other sources than the bird encyclopedia.

The reason for grouping such utterances together is that BirdQuest can never present information to them. Instead, we need to add a number of well-designed responses informing the user on the system's abilities. Utterances that are out of the system's scope require different types of responses from the system, and the corpus gave us valuable insights on the importance of system help messages describing what BirdQuest can and cannot do.

#### 6.1.3 Utterances where BirdQuest fails

Finally, there are those utterances where the system failed, i.e. those where an answer can be found in the encyclopaedia, but where BirdQuest fails to present a successful response for various reasons. Such utterances comprise 27% of the users' input.

We have further analysed these and categorised them as being the result of 1) spelling mistakes, 2) lexical or ontological gaps, or 3) grammatically out of scope, as seen in Table 3. Table 3 includes only utterances that can be successfully responded to, not, for instance, misspellings in utterances that are out of the systems' scope.

Table 3: User utterances not interpreted by BirdQuest

| | No of utterances | Percentage of system utterances |
|---|---|---|
| Misspelled | 28 | 5% |
| Not in lexicon/ontology | 64 | 12% |
| Not in grammar | 50 | 10% |

Table 3 only gives a very brief indication on the nature of non-interpretable utterances in the corpus. For instance, each utterance is tagged as being of one type only, with misspellings having highest priority and missing grammar rules the lowest. Furthermore, there could be several misspellings in one utterance.

It is also the case that the categories overlap, i.e. utterances can belong to more than one category, e.g. a misspelled word can also be missed in the lexicon. There are three such utterances tagged misspelled that also contain words not in the lexicon. 11 of the utterances tagged misspelled cannot be handled because of missing grammar rules. Thus, with a spelling checker, BirdQuest can handle another 14 utterances, i.e. misspelling on its own only accounts for 2.5% of the utterances that BirdQuest cannot handle.

Some interpretation failures are rather complicated and relate to the representation of domain knowledge in the ontology. For example, users tend to ask what colour a specific bird have, see U1 in Figure 21[3]. However, in the bird encyclopaedia colour is linked to the body parts of a bird. The complex relation between the concept "Bird" and the property "Colour" via the body parts, thus, have to be captured in the ontology so that this knowledge can be used for interpretation and dialogue management.

| | |
|---|---|
| U1: | What colour is a blue tit? |
| S2: | Failed to interpret the question. Please try again |

Figure 21: Typical request for a colour.

With more information about how concepts and properties are related, for example, through a chain of hypernym and meronym relations, requests like U1 could be dealt with. Since a bird consists of body parts and body parts have colours the missing link in the request is a body part and based on this a clarification could be produced *A bird has several colours for different body parts. What part of the blue tit are you interested in?* It is, however, also the case that the bird encyclopaedia does not have colour information for all birds. It very much depends on how interesting, or different, the bird is. Often the description relates to other species, as discussed above in Section 4.

# 7 Evaluation of the development process

The BirdQuest ontology and interaction component have been developed iteratively, where running prototype systems have been incrementally extended with more dialogue capabilities following well established steps [11]. The collected dialogue corpus has been the driving force for the iterative work flow. Increments of dialogue functionality have been made simultaneously to both knowledge and data resources, such as the grammars and ontology, and source code functionality of the dialogue system. For this particular domain, we have also experimented with making the utterance representation gradually more elaborate relative the basic model of malin, e.g. by increasing the granularity of the act-type taxonomy and complexity of focal parameter structures, as part of the continual improvement of the malin framework itself and related tools and code repositories. The information extraction component has been used in parallel to extend, and at some points alter, the database content of the final system.

Evaluation of the feasibility of the suggested way of building systems using both information extraction and dialogue, is an important task, but not easily done in practice. We have found that qualitative interviews with developers is a promising approach, both concerning methodology and use of development tools. We have interviewed five developers of BirdQuest about the feasibility of the suggested pre-compiled approach. Three of the developers have been interviewed about source code development and information source engineering, one developer about ontological resource development, and one developer concerning information processing. The interviews showed that both work with the information processing component and the interaction component became closely time-dependent on the design of the ontology. Several interviews indicated that one would like to have the ontology done before the remaining parts, but that in practice the dependencies became iterative in nature. Thus, for a domain where a suitable ontology already exists the rest of the development could be substantially more efficient.

The development of the ontology was in itself relatively straightforward. However, the overlap between the ontology and the lexicon was a source of inconsistency. Several developers expressed

[3] Many of the requests for appearance can be handled by presenting a picture of the bird. However, the pictures in our bird encyclopaedia are copyrighted and can therefore not be presented.

that modelling the domain in terms of objects and properties was not straight-forward in practice. Deciding whether an entity is an object or a property must facilitate both management of focal inheritance and the SQL data lookup. As a consequence, a large part of the practical development work was concerned with making the SQL templates robust and complete with respect to possible user dialogue moves. A suggested improvement concerned details of the internal representation of the dialogue moves, in order to facilitate this. Relational expressions, e.g. requests concerning comparisons between different bird species, were also perceived as hard to represent within the current framework. It was difficult to change the knowledge representation due to its dependencies between information extraction, the interaction component, and the ontology.

For the interaction component, most of the development effort was on task handling. This was the phase most closely related to the ontology and the domain. Consequently less work was on pragmatic interpretation. However, these two phases were perceived by the developers to be closely related in practice, and the pragmatic interpretation tended to depend on prior work of the task handling: "decide how to handle complete user requests before handling ones that arise through dialogue". This made the interaction component instable at times, and decreased the system's performance to some extent.

The iterative dependencies between the two interdependent architectural components also made testing crucial, cf. [4]. In the current implementation testing was done manually, using simple handwritten test code at best. Testing for backwards compatibility had to be done both with respect to context of dialogue and conceptual taxonomy. Newly added functionality easily (re-)introduced faults in previously working code, in one of these dimensions. This was also perceived as originating from the iterative strategy of how to extend the system incrementally by adding dialogue abilities based on a dialogue corpus.

To sum up, the design of the database access, model and contents becomes an elaborate endeavour when information extraction is introduced for interaction with dialogue. Thus, development related to the database became more iterative in nature, which is an overhead that one often tries to avoid due to its inherent complexity. This also suggests one of the limitations for a pre-compiled approach - at some point manual incremental changes of the database simply makes adjustment of the interaction component a much too expensive enterprise, in terms of both man hours and maintenance of system correctness. At that point one should probably search for an alternative or complementary method. But in cases where the domain, or parts of it, is fixed and ontological structures exist in advance of the project, the pre-compiled approach will work effectively.

# 8 Summary

We have presented BirdQuest, a system that utilises dialogue interaction to help users formulate requests for information that has been automatically extracted from a text book. Results from an evaluation of BirdQuest showed that it is possible to develop such a system, and that users get a fair amount of information that they would not have been able to get if the system did not allow for connected dialogue and used a domain ontology. The evaluation also revealed a number of issues for further development of this type of systems.

A more fine-grained ontology with a more complex representation would make it possible to make more inferences and detect information with more details. This is crucial both for the information extraction component and dialogue management. Combining general and core ontologies efficiently with domain ontologies is also important for rapid application development.

More knowledge on the relationship between question types and answer types can guide the information extraction as well as aiding the dialogue management, cf. [31]. The problem of collecting user questions that will guide the design of dialogue systems must however be dealt with. Factoid questions are relatively easy to collect from real usage and re-apply in Q&A systems. Information systems with dialogue capabilities require empirical data containing connected dialogue.

A more sophisticated generator that uses ontological knowledge is also desirable. Another issue is to generalise the knowledge sources, e.g. creating a general question type taxonomy and incorporating other general ontologies like WordNet.

It is not certain that every domain need connected dialogue to the same extent. In complex applications such as the tax domain or legal documents concerning salary settlements, many different factors contribute to the solution to a specific user problem and thereby require clarifications, additional information, resolution of ambiguities, etc. Other application areas such as encyclopaedia lookup may need less dialogue features.

Moving between domains and tasks require customisation, in most cases more than you want. Better tools to make the process of building extraction patterns more efficient must be developed. Several researchers have pointed out this within the IE framework, cf. [30], but in advanced Q&A and dialogue systems, the problem is to identify the user tasks in relation to the document base. Distilling important user tasks from a question corpus is complicated. And yet, the users and the texts are the ones that set the limits and possibilities of the system.

BirdQuest was developed for a closed domain and future research includes the move to multi-domain and then to open domain applications. It is still very much an open question to what extent the techniques for IE, shared knowledge sources and dialogue management presented in this article can be applied for such applications An interesting mixed approach is also to use a simpler information extraction technique as fall-back strategy when the pre-compiled approach fails.

## 9 Acknowledgement

This research is financed by Vinnova, Swedish Agency for Innovation Systems and SITI, The Swedish Research Institute for Information Technology.

## References

[1] Lars Ahrenberg, Magnus Merkel, and Mikael Andersson. A system for incremental and interactive word linking. In *Proceedings from The Third International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas*, 2002.

[2] Jan Alexandersson and Norbert Reithinger. Designing the dialogue component in a speech translation system. In *Proceedings of the Ninth Twente Workshop on Language Technology (TWLT-9)*, pages 35–43, 1995.

[3] James F. Allen, Donna K. Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. Toward conversational human-computer interaction. *AI Magazine*, 22(4):27–37, 2001.

[4] Kent Beck. *eXtreme Programming explained*. Addison-Wesley, 2000.

[5] S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. Dialog in the RAILTEL telephone-based system. In *Proceedings of International Conference on Spoken Language Processing, ICSLP'96*, volume 1, pages 550–553, Philadelphia, USA, October 1996.

[6] Nicole Beringer, Ute Kartal, Katerina Louka, Florian Schiel, and Uli Türk. Promise - a procedure for multimodal interactive system evaluation. In *Proceedings of the Workshop 'Multimodal Resources and Multimodal Systems Evaluation'. Las Palmas, Gran Canaria, Spain*, 2002.

[7] J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C.Ỹ. Lin, S. Maiorano, G. Miller, D. Moldovan, B. Ogden, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski, E. Voorhees, and R. Weishedel. Issues, tasks and program structures to roadmap research in question & answering (q&a). http://wwwnlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc, 2001.

[8] C. Cardie. Empirical methods in information extraction. *AI Magazine*, 18(3):65–80, 1997.

[9] Jennifer Chu-Carroll. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Proceedings of 6th Applied Natural Language Processing Conference*, pages 97–104, 2000.

[10] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of oz studies – why and how. *Knowledge-Based Systems*, 6(4):258–266, 1993. Also in: *Readings in Intelligent User Interfaces*, Mark Maybury & Wolfgang Wahlster (eds), Morgan Kaufmann, 1998.

[11] Lars Degerstedt and Arne Jönsson. A method for systematic implementation of dialogue management. In *Workshop notes from the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Seattle, WA*, 2001.

[12] Matthias Denecke. An information-based approach for guiding multi-modal human-computer-interaction. In *IJCAI´97, Nagoya, Japan*, pages 1036–1041, 1997.

[13] Annika Flycht-Eriksson and Arne Jönsson. Some empirical findings on dialogue management and domain ontologies in dialogue systems – implications from an evaluation of birdquest. In *Proceedings of 4th SIGdial Workshop on Discourse and Dialogue, Sapporo, Japan*, 2003.

[14] T.R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5:199–220, 1993.

[15] Joakim Gustafson and Linda Bell. Speech technology on trial: Experiences from the august system. *Natural Language Engineering*, 6(3-4):273–286, 2000.

[16] Eli Hagen. An approach to mixed initiative spoken information retrieval dialogue. *User modeling and User-Adapted Interaction*, 9(1-2):167–213, 1999.

[17] P. Heisterkamp, S. McGlashan, and N. Youd. Dialogue semantics for a spoken dialogue system. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP'92*, Banff, Canada, 1992.

[18] Kristiina Jokinen, Hideki Tanaka, and Akio Yokoo. Context management with topics for spoken dialogue systems. In *Proceedings of the 36th Annual Meeting of the Association of Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL'98, Montreal*, pages 631–637, 1998.

[19] Arne Jönsson. Dialogue actions for natural language interfaces. In *Proceedings of IJCAI-95, Montréal, Canada*, 1995.

[20] Renaud Leceuche, Dave Robertson, Catherine Barry, and Chris Mellish. Evaluating focus theories for dialogue management. *International Journal on Human-Computer Studies*, 52:23–76, 2000.

[21] K. Mahesh. Ontology development for machine translation: Ideology and methodology. Computing research laboratory mccs-96-292, New Mexico State University, 1996.

[22] K. Mahesh and S. Nirenburg. A situated ontology for practical NLP. In *Proceedings of IJCA'95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995.

[23] D. Moldovan, S. Harabagiu, R. Girju, P. Morarescu, F. Lacatusu, A. Novischi, A. Badulescu, and O. Bolohan. Lcc tools for question answering. In *Proceedings of The Eleventh Text Retrieval Conference (TREC 2002)*. NIST, 2002.

[24] Charles Rich and Candace L. Sidner. Segmented interaction history in a collaborative interface agent. In *Proceedings of the International Conference on Intelligent User Interfaces, Orlando, Florida*, pages 23–30, 1997.

[25] J.F. Sowa. Distinctions, combinations, and constraints. In *Proceedings of ICCS'95 Workshop on Basic Ontological Issues in Knowledge Sharing.*, Montreal, Canada, 19–20 August 1995.

[26] Roland Staav and Thord Fransson. *Nordens fåglar.* Stockhom: Norstedt, 1991.

[27] P. Tapanainen and T. Järvinen. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71, 1997.

[28] Ellen M. Voorhees. Overview of the trec question answering track. In *Proceedings of The Eleventh Text Retrieval Conference (TREC 2002).* NIST, 2002.

[29] Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. Paradise: A framework for evaluating spoken dialogue agents. In Mark Maybury & Wolfgang Wahlster, editor, *Readings in Intelligent User Interfaces.* Morgan Kaufmann, 1998.

[30] Roman Yangarber and Ralph Grishman. Customization of information extraction systems. In *Proceedings of International Workshop on Lexically Driven Information Extraction; Frascati, Italy*, 1997.

[31] Ingrid Zukerman and Eric Horvitz. Using machine learning techniques to interpret wh-questions. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France*, 2001.