Design of Ontologies for Dialogue Interaction and Information Extraction

Annika Flycht-Eriksson
Department of Computer and Information Science
Linköping University, SE-581 83, LINKÖPING, SWEDEN
annfl@ida.liu.se

Abstract

In this paper we discuss how information extraction can be integrated in a dialogue system, with the focus on ontologies as domain knowledge sources. The requirements of domain ontologies are analysed in light of the tasks performed in the two areas, and implications for the design and construction of domain ontologies are presented. One of the most important features of a domain ontology to be used for both dialogue interaction and information extraction is that it should incorporate both the users' conceptualisation of the domain and the conceptualisation present in the information sources. It is shown how this can be done and with the constructed ontology as a starting point more detailed design issues are considered.

1 Introduction

As more and more information has been made available in unstructured and semistructured text format the demands on support for users to seek and retrieve information have increased. A future direction is to use dialogue instead of isolated questions in Question Answering systems (Q/A) [18], thus bringing together the areas of dialogue system research and information extraction. Combining text document processing techniques from IE and Q/A with the type of dialogue interaction provided by dialogue systems involves many challenges. One is the knowledge sources such a combined system would need to represent domain knowledge. Ontologies as means of representing and supporting reasoning about domain knowledge are becoming increasingly common, and are used in several NLP-systems today, for example, Q/A-system (cf. [16; 33]) and knowledge-based machine translation (cf. [24])

The purpose of this paper is to investigate what functions ontologies used for dialogue interaction and information extraction should support, what implications these have on the design of such ontologies, and if it would be futile to combine them into one shared knowledge source to support the integration of information extraction in dialogue systems.

The paper is organised as follows. In section 2 the term ontology is clarified. Section 3 presents a possible architecture for combination of information extraction and dialogue interaction. In section 4 general issues concerning design of ontologies are introduced and discussed. Section 5 briefly presents the dialogue system BIRDQUEST, and in section 6 it is described how the ontology for BIRDQUEST was constructed. Finally, in section 7 more specific design issues are discussed based on the requirements posed by BIRDQUEST and other dialogue and information extraction systems. The paper is concluded by a summary and future directions in section 8.

2 Ontologies

The term ontology is used very differently in various areas of computer science. A general and commonly used definition given by Gruber [13] is that

> An ontology is a *formal explicit* specification of a *shared conceptualisation*.

The keyword in this definition is *conceptualisation*. A conceptualisation is an abstract simplified view of a domain, it identifies the concepts relevant in representing the domain. This view should reflect consensual knowledge and thus be *shared* by a group. An ontology describes this conceptualisation by making the concepts and relations *explicit*, i.e. by defining terms and axioms, in some *formal* language that is machine readable.

An ontology is thus not a natural generic representation of the world that can be discovered and formalised. An ontology to be used for practical NLP is constructed for a specific situation. A definition of a *situated ontology* is given by Mahesh & Nirenburg [25]:

> A situated ontology is as a world model used as a computational resource for solving a particular set of problems.

They consider an ontology as a database with information about what categories (or concepts) exist in the world/domain, what properties they have, and how they are related to one another. Depending on the purpose of the ontology, i.e. the tasks it is used for in a system, this information can be modelled in very different ways. In other words, the design of an ontology is highly dependent on its intended function.

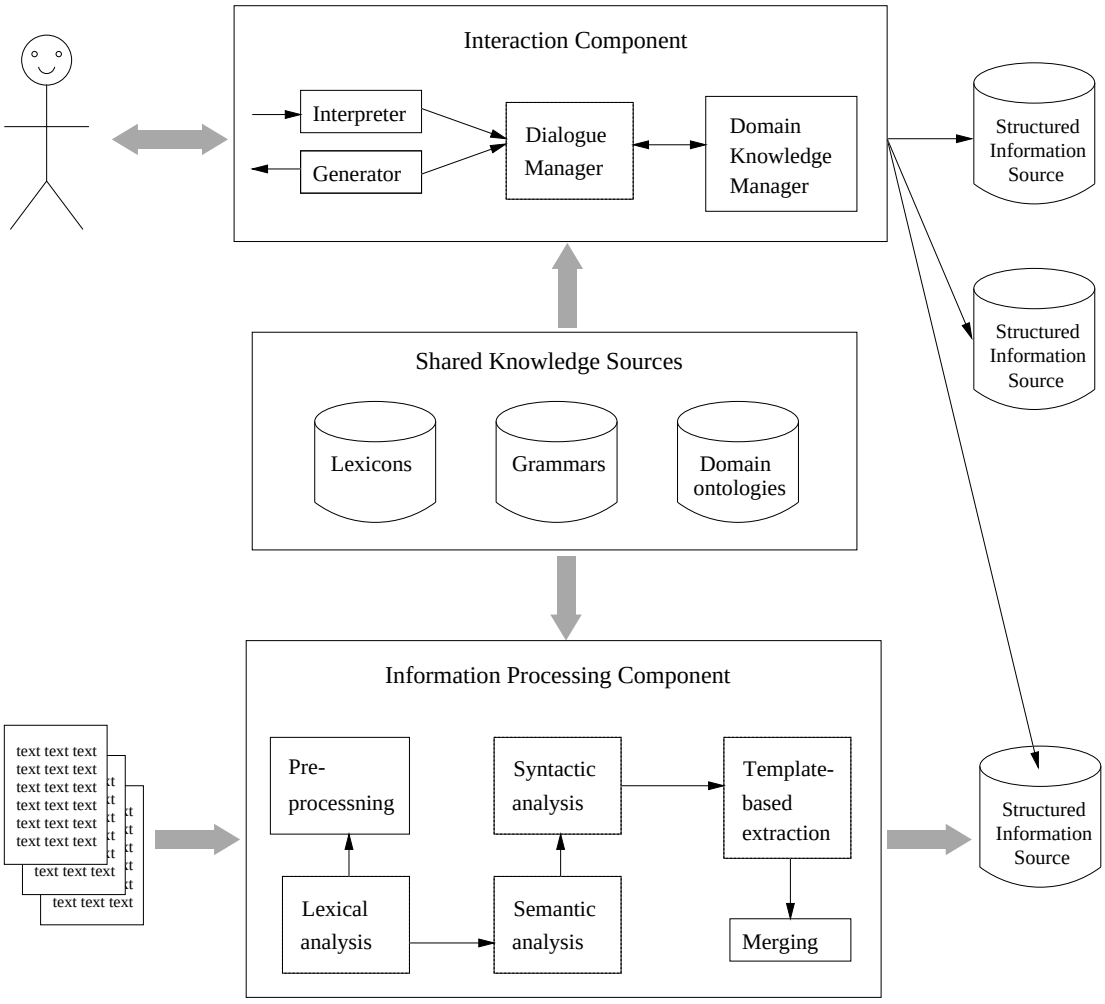

Figure 1: Architecture for a system that combines information extraction with dialogue interaction. The figure shows the different components and the shared knowledge sources used by the components.

3 Combining dialogue interaction with information extraction

Combining dialogue interaction with information extraction has several benefits; dialogue is a natural and efficient means of interaction and with information extraction techniques information can be retrieved from unstructured information sources that are otherwise hard to manage and search for a user. A possible way of merging these two in a practical system is to have two components, an information processing component and an interaction component that as a basis for their tasks use a set of knowledge sources that define the scope of the language and domain. See figure 1 for a possible architecture.

The Information Processing Component takes collections of unstructured or semistructured documents and transforms them into structured information that can be used by the Interaction Component during interaction with the user. 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 query and access the structured information sources to retrieve an answer to the query, i.e. it performs the task of an information-providing dialogue system.

The knowledge sources comprise lexicons, grammars and domain ontologies. Building lexicons and grammars to be used for different tasks involves several challenges but will not be further discussed in this paper. An ontology represents the concepts present in a given domain and how they are related in this domain. Thus, ontologies provide a common vocabulary that can be used to state facts and formulate questions about the domain. Constructing an ontology that can be shared by the Information Processing Component and the Interaction Component would then provide a means for bridging the gap between user requests and the information in the unstructured documents.

4 Ontology design

Designing an ontology involves several decisions on various levels of detail. Noy & Haffner [27] have presented a framework for comparison of ontology design focusing on both general issues as well as more detailed issues concerning representation of content. Taking this framework as a starting point we will discuss the requirements dialogue interaction and information extraction place on an ontology and how these affect the design.

The general design issues concern the purpose of the ontology and basic decisions about the type of knowledge to model and how it should be organised. In figure 2 these have been formulated as questions.

1. What is the purpose of the ontology?
2. What is the coverage of the ontology?
3. What type of concepts should be represented?
4. What top level divisions should there be?

Figure 2: Questions reflecting general design decisions that should be considered before the construction of an ontology.

4.1 Purpose

Noy & Haffner [27] give examples of four different purposes for ontology creation; natural language applications, theoretical investigations, knowledge sharing and reuse, and simulation and modelling. To represent knowledge used in a system with dialogue interaction and information extraction clearly falls within the first category. However, as pointed out in [2], even within an NLP-application an ontology is expected to fulfill various functions such as:

- organizing world knowledge
- organizing the world itself
- organizing 'meaning' or 'semantics' of natural language expressions
- providing an interface between system external components, domain models, etc. and NLP linguistic components
- ensuring expressibility of input expressions
- supporting the construction of 'conceptual dictionaries'

To organise world knowledge is the general function of ontologies and it is also the primary function for ontologies used for information extraction. Most IE-systems use extraction templates based on events that specify entities and their relations in terms of their participation in the event. The ontology should therefore capture and organise the relevant objects, events, properties and their relations. An example of this can be found in the LaSIE system [12] where the domain of business meetings is captured in an ontology that contains objects such as persons, places and organisations, and events involving persons attending meetings. Attributes express the properties of objects and events, as well as the relation between them. This ontological knowledge is used to provide presuppositions and support for coreference resolution and inferences, which can be used for the construction of a template from extracted information pieces.

A secondary function of ontologies used for information extraction is to organise the semantics of natural language expressions, i.e. the source text. In the LaSIE system [12] the text is interpreted into quasi-logical forms (QLF) that are conjunctions of first order logical terms. With the use of the ontology the QLFs can be interpreted and merged into a discourse model.

The functions of ontologies for dialogue interaction are very similar. Since a crucial task of dialogue interaction is to correctly interpret user requests in a dialogue context, important functions of the ontology are to organise the semantics of natural language expressions and knowledge of the world as perceived by the user. Besides these, the ontology also has the function of providing an interface to external information sources from which the system collects answers to user requests. For example, in the Q/A-system FALCON [16] the world knowledge held by an ontology is used to decide on an expected answer type for an information request. The question "What is the wingspan of a condor?" results in the answer-type dimension since wingspan is a hypernym of dimension. The answer type can then be mapped to a named entity category that is used for retrieval of the information, for example, dimension is mapped to quantity.

Thus for a dialogue system that incorporates information extraction the functions of the ontology are to organise world knowledge, organise 'meaning' or 'semantics' of natural language expressions, e.g. user utterances or source texts, and provide an interface between system external components, domain models, etc. and NLP linguistic components.

4.2 Coverage

Coverage of an ontology can range from general commonsense knowledge to detailed application-specific knowledge. Guarino [14] distinguish four kinds of ontologies:

Top-level ontologies, which include general concepts like time and space, objects, events etc. This type of ontology is domain-independent and should therefore be applicable for all problems and applications.

Domain and Task ontologies, which capture, respectively, a generic domain or a generic task. Either of these types can be constructed by specification of concepts in a top-level ontology.

Application ontologies, which are both domain and task specific. These can be constructed through specification of a set of domain and task ontologies related to the application.

As exemplified by the ontology in the LaSIE system [12] ontologies for information extraction are often very application-specific, representing aspects of a few specific events that correspond to the extraction templates, and are thus restricted in both domain and task. In a similar way ontologies used in dialogue systems are closely linked to the task and domain at hand, for example booking a meeting [28].

The aim of Q/A-systems is to be able to answer various types of questions in open domains. Thus, ontologies designed for this type of application need to be very general and cover all types of entities and phenomena, i.e. they need to be generic domain ontologies. For example, to answer "why"-questions the ontology is required to represent causality. However, today's Q/A-systems are limited to factual questions [18] and the ontologies used are often object taxonomies that are created to cover the TREC-corpora thus being more multi-domain ontologies with very varying coverage for the different areas, or more general lexical resources like WordNet [26].

A goal with systems that combine dialogue interaction and information extraction is to provide better access to unstructured information, ultimately in open domains. However, that would require a gigantic amount of work on development of suitable knowledge sources. Attempts to build such ontologies, cf. CYC [23] and WordNet [26], have shown the complexity involved and the problem of finding the right level of detail to be practically useful.

Thus, a more feasible approach seems to be to begin with systems that work within a limited number of domains using smaller and more specialised domain and task ontologies. The coverage can then be stepwise extended by addition of more ontologies, given that a framework that facilitate integration of new ontologies is used, i.e. that the same type of top-level ontology is used as a basis for development of the task and domain ontologies.

4.3 Type of concepts

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 is event and process. In some cases these are seen as a subclass of entities but often they are distinguished from objects, (cf. Sowas ontology [29] and Mikrokosmos [24]).

To define and describe these two types of concept, properties and relations are used. They are often implicit, for example, seen as part of a thing's internal structure, e.g. 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; 3]). 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. [4]).

Although explicit representation of properties and relations is not as common as the other two types of concepts, they do exist. In Mikrokosmos [25], properties, which include attributes and relations, are represented both as explicit concepts and slots in objects and events.

Based on formal ontology Guarino [15] argues that to avoid semantic confusion a clear distinction should be made between *particulars* and *universals*, which roughly can be seen as corresponding to the entities of the world and the properties and relations used to describe them. He proposes two separate taxonomies for these.

Explicit representation of properties and relations can also be motivated from a dialogue management perspective, since it facilitates contextual interpretation of requests and aid clarifications, for example, in the LINLIN-architecture [21] objects and properties are the basis for modelling focal information, and for interpretation of anaphora and ellipsis.

Thus, it seems that for the combination of dialogue interaction and information extraction, explicit representation of all four types of concept should be present in the ontology. Events and objects are primary for both information extraction and dialogue while properties and relations are useful mainly for dialogue system.

4.4 Top level divisions

As already stated in section 4.2 it should be possible to construct a generic upper top-level ontology that could be used as a unifying framework for all possible domains and tasks. However, in practice there exist several different top-level ontologies. Some distinctions are very common, for example, abstract vs concrete, and exist in almost all, while others are more unusual.

From a pragmatic point of view it is possible to build application ontologies for dialogue interaction and information extraction without these distinctions. But if existing ontologies are to be integrated with new ontologies, a common top-level plays a crucial part in integrating and mapping concepts.

Ontologies built for the purpose of dialogue interaction must be designed to incorporate the user's view of the world, or part of the world, in terms of the types of entities that can be included and how they can be organised. The choice of top-level distinctions should therefore be linked to language and cognition. Gangemi et al. [11] discuss how this can be accomplished. They argue that whilst there may be certain differences between different cultures, languages, etc, some parts depend on common cognitive structures due to humans sensorial interaction with reality. A methodology for the design of top-level distinctions based on this assumption is outlined and a number of distinctions are proposed based on this. These include concrete versus abstract, and individual versus collection, which are the most common top level divisions present in existing ontologies [27], but also dependence versus independence, and extensionality.

Thus, when designing an ontology to be used in a dialogue system one should consider the importance of reusability and extendibility. To be able to integrate new domains or tasks the top-level distinctions must be general to a certain extent.

5 BIRDQUEST

To get more insights into the possibilities and challenges of designing ontologies that could act as shared knowledge sources in systems combining dialogue interaction and information extraction an ontology was constructed for a specific application.

Based on the architecture proposed in section 3, a system called BIRDQUEST has been developed [22]. It provides users with information about birds through natural language dialogue. An example dialogue that illustrates what the system can and cannot do is presented in figure 3.

The information provided is collected from a structured information source, a database, which is filled with information extracted from a bird encyclopedia by the Information

U1: what does an ostrich look like
S2: Information about non-nordic birds is missing.
The available birds are listed in the menu ”Birds a-ö”. Please try again
U3: What colour are swans?
S4: The system failed to interpret the question. Please try again.
U5: Where do swans live?
S6: Choose one of the following swans: song swan, whooper swan, mute swan.
U7: Where do mute swans live?
S8: Information about a mute swan’s range:
The mute swan can be found nesting mainly in western Europe, but also in southeast Europe and Turkey.
U9: How much does it weigh?
S10: Failed to interpret the question. Please try again.
U11: For how long does a mute swan live?
S12: Information about maximum age of a mute swan:
Maximum age in the wild: 24 years and 3 months
U13: what does a mute swan look like
S14: You can receive information about the size and plumage of a mute swan. Please specify what you are interested in.
U15: plumage
S16: Information about eclipse plumage is missing for Mute Swan.
Information about winter plumage is missing for Mute Swan.
Information about summer plumage is missing for Mute Swan.
Information about breeding plumage is missing for Mute Swan.

Figure 3: A dialogue illustrating the BIRDQUEST system.

Processing Component. A wide variety of pattern extractor rules are used to identify the relevant information as slots and fillers. The objective is to fill the database with relevant information and ignore text segments that do not meet the needs of the users. Figure 4 illustrates how unstructured text is transformed into slot and filler type information in the database.

Original text

```
Black-throated diver
Gavia arctica
58-73 cm, wingspan 110-130 cm.
Somewhat larger than the
red-throated diver with wider
neck and straight, dagger-
shaped beak
```

Extracted information

NAME:	Black-throated diver
LATIN_NAME:	Gavia arctica
MAX_WING:	130
MIN_WING:	110
MAX_HEIGHT:	73
MIN_HEIGHT:	58

Figure 4: Original text passage from the text book and the corresponding entry in the database (translated from Swedish).

The interaction component in BIRDQUEST is based on the MALIN framework [7]. MALIN is a modularised dialogue system that particularly it separates dialogue management (DM) from domain knowledge management (DKM) [9]. The former handles the dialogue whereas the latter handles access to various background information sources.

The Dialogue Manager is responsible for controlling the flow of the dialogue by deciding how the system should respond to a user utterance. This is done by inspecting and contextually specifying the information structure produced by an interpretation module. In MALIN, dialogue history is represented in dialogue objects with a parameter termed Objects which identifies a set of primary referents and Properties which denote a complex predicate ascribed to this set [20]. In BIRDQUEST Objects are normally birds and Properties model information about the birds, such as appearance, number of eggs and feed.

The DKM receives requests from the DM and processes them further using domain knowledge, for example, disambiguation and mapping of vague concepts to ones more suitable for database access. It then retrieves and coordinates information from available information sources, such as data and knowledge bases. If a request is under-specified or contains inconsistencies from the DKM’s point of view, a specification of what clarifying information is needed will be returned to the Dialogue Manager to help the formulation of a clarification question to the user.

6 Ontology construction

There are no standard methods for development of ontologies. ”Building ontologies is still a matter of craft rather than an understood engineering process” [19, p. 13]. However, a number of stages in the development process are usually passed through during construction of an ontology:

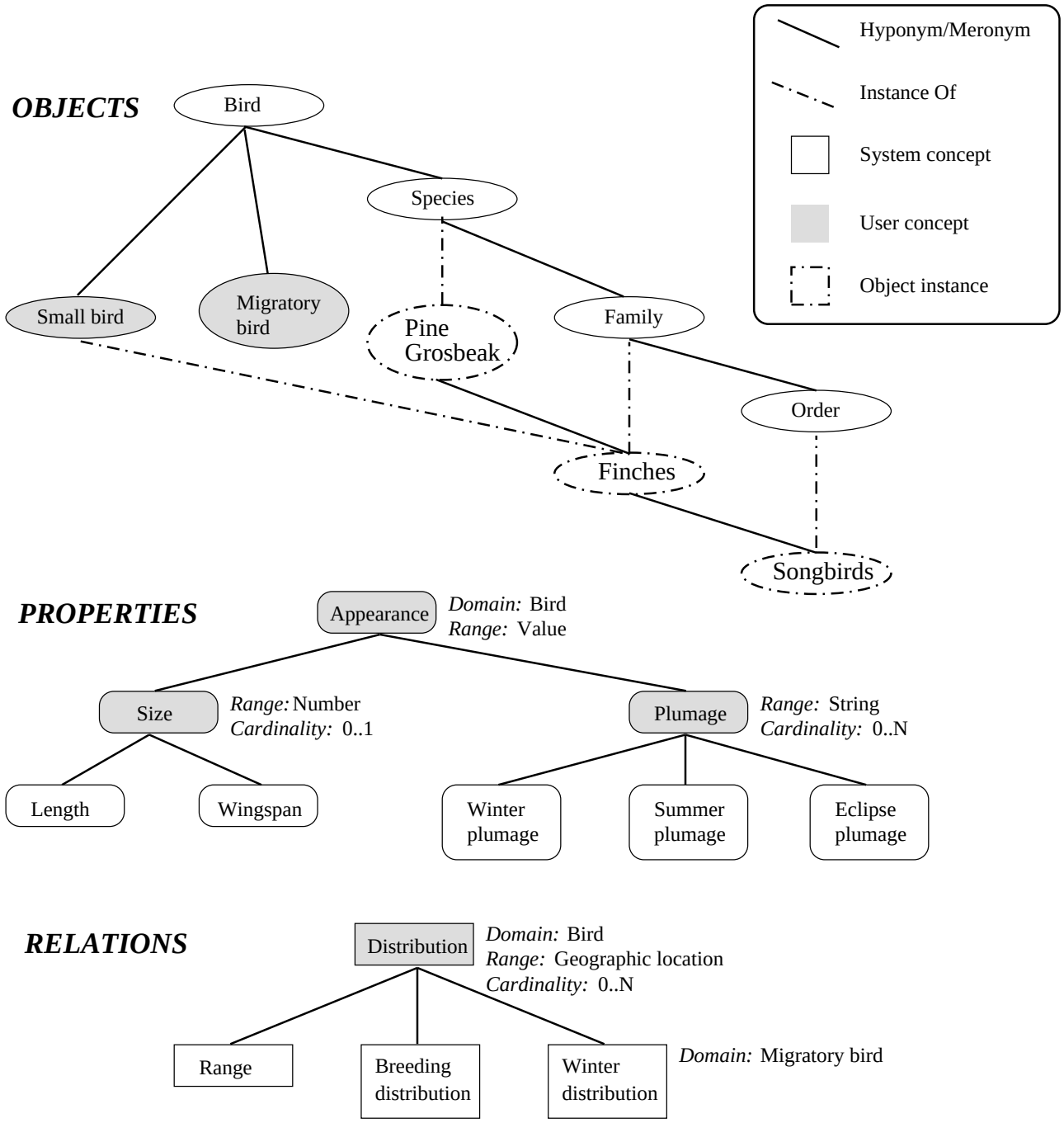

Figure 5: Part of the integrated ontology representing the conceptualisations of both bird encyclopedia and users.

1. Specification of scope and purpose
2. Data collection
3. Conceptualisation and organisation
4. Formalisation and implementation
5. Evaluation

6.1 Scope and purpose

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. It is worth pointing out that an ontology to be used in a NLP-

system is a computational resource used for a specific purpose and therefore not an objective representation of the world, but rather a very subjective view of the world. In the case of a combined system for dialogue interaction and information extraction it is crucial that the ontology captures both the system-oriented view to reflect the information sources and the user-oriented, often more naive, view of the domain.

The bird encyclopedia almost exclusively contained information of a factual character with focus on objects. The ontology was therefore restricted to modelling this type of information in terms of objects, properties and relations, leaving out events/processes.

6.2 Data collection

To identify relevant concepts to include in the ontology two corpora were analysed, the bird encyclopedia [30] and a question corpus.

The bird encyclopedia

The organisation and structure of the book were taken as a starting point for identification of 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, properties and relations relevant for the purpose of information extraction. The analysis gave a total of 8 categories (various groupings of birds, and geographical locations), 30 properties and 3 relations, some of which are presented in figure 5.

The question corpus

The corpus used 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 properties were identified. These new concepts were of three types:

- Users sometimes utilised another way of categorising birds than the biologically oriented taxonomy in the bird encyclopedia, talking about ”Spring birds”, ”Small birds”, ”Migratory birds”, and ”Birds of prey”, etc.
- In many cases the properties of the birds were more general than the terms used in the book, for example questions about size which includes both wingspan and length.
- A number of properties were not present in the bird encyclopedia but closely related to them, such as weight and speed of flight.

6.3 Conceptualisation and organisation

From the analysis of the encyclopedia a conceptualisation underlying the structure and presentation of information that was to be extracted by the Information Processing Component was constructed. The result was a system-oriented domain ontology representing experts' (the book authors) view of the domain. The non-expert view of the domain useful for dialogue interaction as provided by the question corpus were then integrated in the following manner:

- By allowing multiple inheritance new links between existing categories and new categories were added. Note, for example, how the category ”Small bird” is introduced and that ”Finches” is multiple linked to both this category and ”Families” in figure 5.
- In a similar manner the vague properties were introduced and linked to the existing properties. Figure 5 illustrates how two new levels are introduced, ”Wingspan” and ”Length” are subproperties of ”Size”, which in turn is a subproperty of ”Appearance”.

Properties from the question corpus which could not be linked through hyponym relations to existing properties in the system-oriented conceptualisation where left out. This was because there was no existing information regarding these in the bird encyclopedia and hence nor was there in the database to be used by the dialogue system.

With 4 new objects and 6 new properties introduced by the analysis of the question corpus, the resulting ontology contained 12 objects, 36 properties and 3 relations.

6.4 Formalisation and implementation

The ontological knowledge was represented in two different knowledge bases, 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. Properties and Relations also had domain and range restrictions, which stated what type of objects they were applicable to. This is illustrated in figure 5 where the relation “winter distribution” has domain “Migratory bird” and range “Geographical location”. There were also cardinality restriction for the properties and relations.

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 some species the migratory behaviour varies for young and old birds. Thus only ”Birds of prey” had instances in the implementation.

6.5 Evaluation

To verify that the ontology met its intended purpose it was evaluated in the context of its usefulness for dialogue interaction [10].

A dialogue corpus was collected during a day when the public was invited to the university. It was a walk-up and use situation where the users received minimal instructions

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 has a length of 56 cm from beak tip to tail.
Information about wingspan is missing for Gadwall

Figure 6: A dialogue excerpt illustrating a successful information exchange where ontological knowledge is used to guide clarification requests and database access.

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. Approximately half of these (244) were involved in successful information exchanges were the user initiated a request and the system, often after clarification sub-dialogues guided by ontological knowledge, retrieved the correct answer from the database. An example excerpt is shown in figure 6.

The basis for this type of clarification is domain knowledge collected from the domain ontology. Utterance U17 is underspecified since the object, the bird family "Waterfowl", can refer to a number of different species of bird, and the property "Appearance", is vague. To pose clarification question S18, information about which species belong to the given family is gathered from the part of the ontology that contains facts and the user is asked to choose one of them. Next, in S20, the concept part of 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 first type of clarification where the object needs to be specialised were initiated in 31 of 180 user requests for information. The second type regarding vague user properties occurred in 28 instances. Finally, mapping of vague properties to ones suitable for database access was done in 64 cases. There were also 5 cases in which the user requested information about birds of the categories "Small birds", "Migratory birds" and "Sedentary birds" which had not been implemented and could therefore not be handled. The high number of clarifications and mapping of properties for database access shows the usefulness of the domain ontology.

7 Requirements and design issues

During the work with the ontology in BIRDQUEST a number of more detailed design issues were encountered. Based on this experience a more fine-grained analysis of requirements and design of ontology content was conducted. Figure 7 presents a list of design decisions based on the frameworks by [5; 27]. For each issue it is indicated if it is a requirement for dialogue interaction or information extraction.

7.1 Concepts and taxonomy

Concepts in ontology can vary from nearly atomic to highly structured. Mahesh & Nirenburg [25] advocate the latter. They mean that it is the rich *inner structure* that allows for sophisticated use of the ontology, for example, to perform disambiguation. However, if the properties of the concept and relations to other concepts represented as slots in a frame instead are made explicit, i.e. properties and relations are represented separately, the same functionality would be available with less complex concepts. Thus concepts do not need to have internal structure as long as information about properties and relations is maintained elsewhere. This is illustrated in the BIRDQUEST ontology that has atomic concepts for objects and separate taxonomies for properties and relations.

The ability to define *metaclasses*, i.e. classes as instances of other classes, and the *Subclass_Of* relation between two classes is the foundation for constructing a basic IS_A taxonomy. This type of taxonomy is necessary for many dialogue interaction and information extraction tasks, for example coreference resolution [6]. Since a concept in one expression might be a hyponym or hypernym of a concept in another expression the Subclass_Of relation is useful. Methods for merging, or unifying, partial results produced by information extraction also rely on this type of taxonomic knowledge.

Knowledge about hyponym relations can also be used for the creation of extraction patterns. For example, in [1] it is shown how specific extraction rules generated from training data can be generalised by replacing concepts with their super-ordinates using the hyponym/hypernym relations in WordNet. For example, IBM Corporation can be replaced by {business, concern} which in turn can be generalised to {enterprise} and so on.

For dialogue interaction the taxonomic knowledge can be used to handle clarifications. In the case of the ambiguous question "What is the biggest bird?" a clarification of the sought property "Do you mean in terms of length or wingspan?" can be produced. Thus, to handle clarification sub-dialogues it is important that the ontology contains a taxonomy not only over the different categories of objects, but also the properties.

Taxonomies can be organised in untangled tree structures where a concept is only allowed one hyponym category. Others based on a distinction approach, rely heavily on *multiple inheritance*, where new categories are defined through multiple relations to hyponyms. As shown in section 6.3 multiple

Design decision	Dialogue	IE
Concepts and Taxonomy		
Internal structure of concepts	-	-
Metaclasses	x	x
Subclass_Of	x	x
Multiple inheritance	x	?
Part-whole treatment	x	x
Properties and Relations		
Domain restrictions	x	x
Range restriction	x	x
Local	x	x
Instance	x	x
Class	-	U
Polymorph	x	x
Binary relations	x	x
Arbitrary n-ary relations	U	U
Default values	U	-
Cardinality restrictions on values	x	x
Procedural attachments for values	U	U
Instances		
Instances of categories	x	U
Multiple instantiation	x	x
Facts (instances of relations)	x	U
Claims (assertion of fact by instance)	?	?

Figure 7: Requirements placed on design of ontology content for dialogue interaction and information extraction. **x** denotes a requirement, **U** that the feature can be useful, **-** that it is not required and **?** that the role cannot be decided based on present work in the area.

inheritance can be a way to deal with the requirement that an ontology in a dialogue system should reflect both the users' and systems' view of the domain. A category can thus be a subclass of both a user-originated and system-originated category.

There are several types of *part-whole* relations, cf. [32], which are treated to various extents in existing ontologies. Part-whole relations are needed both for dialogue interaction and information extraction. In information extraction they are used for merging of partial results, checking that parts of the information are related by one object being a part of another object. For dialogue interaction, knowledge of part-whole relations can help interpretation of questions, and focus management. For example, in VERBMOBIL [28], the ontology is used to determine if an utterance is a new proposal for a meeting time or a refinement of a previous proposal. The evaluation of BIRDQUEST showed its necessity for clarifications as users tend to ask about the colours birds have (cf. U3 in figure 3). Since colour is linked to the body parts of a bird in the bird encyclopedia the relation between the concept "Bird" and the property "Colour" has to go through a chain of hypernym and meronym relations in the ontology. Using knowledge of these relations, a clarification where the system asks the user for a specific body part may be initiated.

7.2 Properties and Relations

As stated earlier properties and relations can be treated as explicit concepts in the same way as objects and event/processes. *Domain and range restrictions* are then necessary to link properties and relations to the specific type of objects or values they are applicable to. The use of explicit domain and range restriction is useful for both dialogue interaction and information extraction. For dialogue interaction it helps disambiguation of requests, and it can also be used for contextual interpretation of anaphora or ellipsis. For, example if the user in an elliptic utterance provides a property, e.g. "Feed", and there are several objects in focus, e.g. "Seagull" and "Egg", the right one, i.e. seagull, can be chosen based on the domain restriction of the property, i.e. "Feed" is only applicable to birds not eggs. A third use is to formulate clarification requests when an object or value is missing, using the domain and range restriction to indicate what type of information the user should provide.

The most basic type of properties and relations is *local*, i.e. they belong to a specific concept. This is the type of properties and relations commonly found in frame representations used for both information extraction and dialogue systems. When properties and relations are represented separately from the objects this implies that they must have domain and range restrictions that tie them to objects.

The properties can be *instance* oriented, i.e. they allow for different values for each instance of a concept, or *class* oriented, i.e. they have the same value for all instances of a concept. The first type is required for both dialogue interaction and information extraction. It is the value of these that are stored in a database acquired through information extraction on source texts, which is retrieved and presented to the user as a response to an information request. The second

type, class properties, are not necessary but can be useful for information extraction as they define a concept. Thus, they can be used to detect and extract new instances of a concept in order to populate the factual part of the ontology with new instances.

Since we are dealing with natural language interaction within several domains, *polymorph* concepts are bound to occur. This means that several properties and relations can have the same name but be applied to different types of objects and values and should be allowed for in the ontology. A detailed discussion of the role of polysemy in dialogue systems is presented in [28].

In most dialogue and information extraction systems relations are binary and this is also the case in BIRDQUEST. Since it is possible to transform *n-ary relations* to binary, only binary relations are required. However, to formulate extraction patterns and perform template filling n-ary relations might allow representations that are more expressive and easier to use.

Since use of information extraction to populate a database can result in holes in the database, *default values* for properties can be a useful feature. It provides a fall-back during dialogue interaction in cases when the information extraction has failed or information is missing in the sources.

In a dialogue users often use anaphora and ellipsis. This means that fragmentary information must be interpreted in light of the previous dialogue. Thus, the dialogue manager must be able to decide if new information is a complement or refinement of what is already provided and therefore should be integrated with it, or if the focus has shifted, which means that the new information should replace the old. Provided that the ontology had information that restricts the number and type of values different features and properties can take it can be used as a basis to make decisions about how focus should be handled. *Cardinality constraints* can also be used to help the information extraction process populate a database [8].

Procedural attachments to calculate values for a property are not necessary but can be very useful. For example, with information extraction, formulas can be extracted and the correct value calculated based on input to the formula provided by the user through dialogue. They can also be used for information extraction, for example, in [31] demon attributes of three different types, TO-FILL, NORMALIZE and WHEN-FILLED, are used during different stages of template filling.

7.3 Instances

For dialogue interaction it is necessary to include *instances of concepts* in the ontology, although they may be kept in a separate part of the ontology as is done in BIRDQUEST. Since information in the database is kept for certain instances, e.g. bird species, the users' requests for information about more vague or general concepts have to be mapped to the appropriate instance(s). For example, in the question "What do birds of prey eat?" the concept "Birds of prey" is mapped to a set of families of birds and the user can be asked to be more specific by choosing one of the families.

Since ontologies for dialogue systems and information extraction should be able incorporate several views of a domain, e.g. user and system, or several different information sources, *multiple instantiation* is needed. As exemplified in BIRDQUEST, see figure 5, multiple instantiation allows an instance like "Finches" to be an instance of both the user category "Small birds" and a system category "Family".

For dialogue purposes, *facts* about the concept instances, such as the hyponym relation "Pine Grosbeak" IS_A "Finch" and meronym relation "Östergtland" PART_OF "Middle_Sweden" are also required. This information can be used for clarifications in a similar way to facts about the instances of concepts. This is illustrated by S18 in the dialogue in figure 6. Facts can also be useful for information extraction during boot-strapping to create new extraction patterns for similar types of facts.

There is no need to represent *claims* in an ontology used for dialogue systems or information extraction since this type of information is rather captured in the database.

8 Summary and future directions

The design of an ontology is highly dependent on its intended function. A number of general design issues have to be considered before an ontology is constructed for an application. In this paper we have discussed how these affect the design of ontologies that can be used to combine dialogue systems with information extraction.

Based on experience from development of such a combined system, the BIRDQUEST system, a number of more detailed requirements for ontologies to be used for dialogue interaction and information extraction have been identified (figure 7).

In the table some issues are marked **?**. For these, the roles they play for dialogue or information extraction could not be determined based on the work conducted within the area. Further investigation of various applications and domains might reveal their importance. Future work should also include analysis of issues concerning axioms and inference mechanisms which were omitted here.

Another aspect of ontology design that was not included but deserves attention is the representation of space and time. Work with BIRDQUEST indicates that the object taxonomy should include a variety of spatial and temporal objects. These can then participate in relations with other objects or event/processes that have spatio-temporal features. They may also serve as values for constraints on properties and relations, such as "Plumage" or "Feed" that vary with the seasons.

The evaluation of BIRDQUEST also indicated future issues to investigate concerning the use of ontologies in this type of system. For example, since information extraction is used to populate the database with information there can be holes, i.e. for some properties values are missing due to shortcomings in the information extraction component or lack of information in the original text source. This lead to unnecessary clarifications. In U13-16 in the dialogue presented in figure 3, a more appropriate answer to U13 would have been to give information about size instead of making a clarification since there is no information about plumage in the database. Further examples include questions that were outside the database coverage such as "Weight" in U9 in figure 3. If properties similar

to those in the database, such as ”Weight”, ”Flight-speed”, could be added to the ontology as user-oriented properties, the system would have some knowledge of its limitations and could give more informative error messages. If these properties were related to others, for example, ”Weight” is a sub-property of ”Appearance”, the system could even suggest some of the sibling properties, in this case ”Size” and ”Plumage”.

References

[1] A. Bagga, J. Y. Chai, and A. W. Biermann. The role of WordNet in the creation of a trainable message understanding system. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 941–948, Menlo Park, California, 1996.

[2] J. A. Bateman. The theoretical status of ontologies in natural language processing. In *Proceedings of workshops on Text Representation and Domain Modelling - Ideas from Lingustics and AI*, KIT Report 97. Technical University Berlin, 1991.

[3] 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, 1996.

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

[5] O. Corcho and A. Gómez-Pérez. Evaluating knowledge representation and reasoning capabilities of ontology specification languages. In *Proceedings of ECAI-00 Workshop on Applications of Ontologies and Problem-Solving Methods*, Berlin, Germany, 2000.

[6] J. Cowie and Y. Wilks. Information extraction. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*. New York: Marcel Dekker, 2000.

[7] N. Dahlbäck, A. Flycht-Eriksson, A. Jönsson, and P. Qvarfordt. An architecture for natural dialogue systems. In *Proceedings of ESCA Tutorial and Research Workshop on Interactive Dialogue in Multi-modal Systems*, Kloster Irsee, Germany, 1999.

[8] D. W. Embley, D. M. Campbell, S. W. Liddle, and R. D. Smith. Ontology-based extraction and structuring of information from data-rich unstructured documents. In *Proceedings of CIKM'98*, Washington, D.C., USA, 1998.

[9] A. Flycht-Eriksson and A. Jönsson. Dialogue and domain knowledge management in dialogue systems. In *Proceedings of 1st SIGdial Workshop on Discourse and Dialogue*, Hong Kong, 2000.

[10] A. Flycht-Eriksson and A. 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.

[11] A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari. Understanding top-level ontological distinctions. In *Proceedings of IJCAI'01 Workshop on Ontologies and Information Sharing*, Seattle, Washington, USA, 2001.

[12] R. Gazauskas, K. Humphreys, S. Azzam, and Y. Wilks. *Concepticons vs. Lexicons: An Architecture for Multilingual Information Extraction*, volume 1299 of *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, Lecture Notes in Artificial Intelligence*, chapter 3, pages 28–43. Springer, 1997.

[13] Thomas R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. Technical report, Stanford University, Palo Alto, CA, 1993.

[14] N. Guarino. Formal ontology in information systems. In N. Guarino, editor, *Formal Ontology in Information Systems, Proceedings of FOIS'98*, pages 3–15, Trento, Italy, 6-8 June 1998. IOS Press, Amsterdam.

[15] N. Guarino. Some ontological principles for designing upper level lexical resources. In *Proceedings of First International Conference on Language Resources and Evaluation*, pages 527–534, Granada, Spain, 1998.

[16] S. Harabagiu, D. Moldovan, M. Paşca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Gîrju, V. Rus, and P. Morărescu. FALCON: Boosting knowledge for answer engines. In *Proceedings of the Ninth Text REtrieval Conference, TREC-9*, Gaithersburg, Maryland, USA, 2000.

[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] L. Hirshman and R. Gaizauskas. Natural language question answering: the view from here. *Journal of Natural Language Engineering*, 4(7):275–300, 2001.

[19] D. Jones, T. Bench-Capon, and P. Visser. Methodologies for ontology development. In *Proceedings of IT & KNOWS Conference, XV IFIP World Computer Congress*, Budapest, August 1998.

[20] A. Jönsson. Dialogue actions for natural language interfaces. In *Proceedings of IJCAI'95, Montréal, Canada*, 1995.

[21] A. Jönsson. A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering*, 3(2/3):103–122, 1997.

[22] A. Jönsson and M. Merkel. Some issues in dialogue-based question-answering. In *Working Notes from AAAI Spring Symposium, Stanford*, 2003.

[23] Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

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

[25] 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.

[26] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: an on-line lexical database.
URL: ftp.cogsci.princeton.edu/pub/wordnet/5papers.ps, 1993.

[27] N. F. Noy and C. D. Hafner. The state of the art in ontology design, a survey and comparative review. *AI Magazine*, 18(3):53–74, 1997.

[28] J. Quantz, M. Gehrke, U. Kssner, and B. Schmitz. The VERBMOBIL domain model version 1.0. Technical Report 29, Technishe Universität Berlin, 1994.

[29] 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.

[30] R. Staav. *Nordens fglar*. Stockhom: Norstedt, 1991.

[31] L. K. A. Wee, L. C. Tong, and C. L. Tan. Textual information extraction in the face of information deluge. In *Proceedings of IJCAI'99 Workshop Text Mining, Foundations, Techniques and Applications*, Stockholm, Sweden, 1999.

[32] M. E. Winston, R. Chaffin, and D. J. Hermann. A taxonomy for part-whole relations. *Cognitive Science*, 11(4):417–444, 1987.

[33] R. Zajac. Towards ontological question answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, ACL'01, Workshop on Open-Domain Question Answering*, 2001.