Evaluating Focus Theories for Dialogue Management

Evaluating Focus Theories for Dialogue Management

Renaud Lecœuche*†
renaudl@dai.ed.ac.uk

Dave Robertson†
dr@dai.ed.ac.uk

Catherine Barry*
greboval@insa-rouen.fr

Chris Mellish†
chrism@dai.ed.ac.uk

Abstract

Interactive reasoning tools are usually driven by an agenda of tasks to perform, rather than by conventions of human dialogue. On the other hand, theories of dialogue in natural language tend to ignore the constraints imposed by reasoning tools. This paper presents a system composed of a reasoning module and a dialogue manager which cooperate to produce dialogues that are suitable for reasoning and follow human dialogue conventions. The dialogue manager is driven by focus rules. Various competing focus theories exist but there have been few comparative studies of their use in non-trivial tasks. We make a comparative study of the use of focus theories, which requires us to be precise about our interpretation of our chosen focus theories and to develop an innovative means of empirical testing for them. We evaluate the theories on an example of combined dialogue and reasoning from the domain of requirements elicitation.

1 Introduction

Reasoning tools interacting in natural language with their users are usually not aware of rules needed to ensure good dialogues. Questions are asked and statements are made when needed by the task at hand, independently from the dialogue history. This is a problem, since the easiest way to acquire the information for the reasoning tools is often not the most acceptable in terms of human dialogue. Adapting the dialogue, so that users can make sense of the information provided by the tools, is therefore important (Moeller 1996). On the other hand, we do not want to force reasoning tools to be aware of dialogue management strategies. This is all the more important since we may want to re-use the same tool in a different environment with, for example, a graphical or a multi-modal user interface, where rules may differ from those of natural language. A solution to this problem is to separate reasoning tools and dialogue managers, but make them interact and constrain each other.

*PSI-LIRINSA, I.N.S.A., Place Emile Blondel, 76130 Mont-Saint-Aignan, France
†Division of Informatics, 80 South Bridge, Edinburgh EH1 1HN, Scotland

The main aim of this paper is to present a dialogue management system and to show how it can be integrated with a reasoning system to produce dialogues in natural language. The reasoning system imposes some constraints on the dialogue on its own. The dialogue manager interacts with it to provide dialogues that are acceptable for reasoning and follow human dialogue conventions. We have put a particular emphasis on providing coherent dialogues.

Requirements engineering, and especially requirements elicitation, is an interesting domain to evaluate such a system, since reasoning and communication play crucial roles in this process. Requirements elicitation is a difficult part of software engineering in which the specifications for a new system are discussed with potential users. Because verifying that requirements are correct is a complex task, computer support is sometimes beneficial. This support requires formal specifications. However, people are usually not trained to use formal specification languages. Task or domain specific languages may smooth the learning curve when writing formal specifications, but the elicitation process often remains error prone. Users need more support while writing specifications. In particular, a tool which interacts with them and helps them express their requirements in a domain specific way could lower the number of requirements elicitation errors. Allowing a good dialogue seems vital since the very act of expressing requirements is a difficult task.

We do not claim that our approach is a contribution to requirements engineering. First, our system does little work on the requirements themselves. We use and present a simple elicitation module to explain how its reasoning interacts with dialogue management techniques. More appropriate systems such as the Requirements Apprentice (Reubenstein 1990) or KA (Peterson et al. 1994) should be used for more complex tasks. It is also well-known that natural language approaches are not always the most suitable way of expressing requirements. Graphical user interfaces often enable users to carry out simple actions that would require a lengthy dialogue. Finally, other methods could be better adapted for acquiring requirements. Direct elicitation methods are not always suitable (Cooke 1994), since they force people to verbalise their requirements, which may be difficult. Observation and ethnographic methods (Goguen and Linde 1993; Sommerville et al. 1993) may be needed. These approaches cannot be dealt with simply with our system. Our approach may nevertheless provides improvements in the use of some requirements elicitation tools. The Requirements Apprentice is a case in point. Although it allows an incremental and interactive definition of requirements, its communication capabilities are very limited. Our approach offers a way of integrating reasoning and dialogue management, which could improve this aspect. The fact that we emphasize the importance of dialogue is also a departure from existing natural language approaches to requirements elicitation. Most other approaches (Chen 1983; Black 1987; Macias and Pulman 1993; Nanduri and Rugaber 1996; Osborne and MacNish 1996; Moreno 1997; Rolland and Proix 1992; Rolland and Ben Achour 1998) are based on advanced analysis of sentences. However, they ignore dialogue level information. This may make the interaction with the tool less natural than in our approach.

In this paper, we first present in section 2 our approach to dialogue management. Then, we describe in section 3 the architecture of our system. In section 4, we describe in-depth formalisations of the focus theories we use. These formalisations provide rules constraining the dialogue evolution. We present, in section 5, a dialogue example showing how we use the focus theories to organise the dialogue. In section 6, we present an evaluation of our system. We evaluate and compare the theories presented in the paper. We also show that not using dialogue management techniques makes the dialogue barely understandable. In section 7, we describe an innovative method for developing dialogue managers and show how we used this method to develop our program. Finally, we conclude the paper in section 8.

2 Dialogue Management

There are three aspects that are usually taken into account when managing dialogues (Stent 1998): the informational aspect, the intentional aspect (Moore and Paris 1993) and the attentional aspect (Grosz and Sidner 1986). We present them here in an example about the design of World-Wide-Web (WWW) sites. This domain is the application domain of our system. The system has some knowledge of how WWW sites are built and helps a user design his or her own site. We also distinguish for each aspect whether the reasoning module or the dialogue manager is more likely to be able to deal with it.

Example 1

U:	I want a link between the publication page and the home page.
S:	(a) Do you want a link to or from the home page ?
	(b) Let's consider other navigational devices.
	(c) Do you want to link the publication page to other pages ?

The informational aspect deals with the information contained in a message. In example 1 where a user expresses a requirement, the information is that the two pages must be related. We would like at least to ensure that the information given by users is consistent and complete (Davis et al. 1993). In the example, it is not complete. Because links are oriented, we need to know if the link is going from the publication page to the home page or vice-versa. In that case, the system could reply with sentence (a). Since making correctness checks is complex, we believe this should be done by the dedicated elicitation module rather than by the dialogue manager. The dialogue manager will just pass on the information and report back potential problems.

The intentional aspect deals with the intention driving the utterances in the dialogue. In example 1, the intention might be to speak about navigation through the pages. In that case, the system could provide guidance by proposing, as in sentence (b), to speak about other navigational devices such as a navigator or site map. However, it is not clear whether taking into account intentions is a sustainable approach. First, recognising users' intentions is a difficult task (Cohen et al. 1990). Second, the intentions driving the system are often left implicit in its design. There is usually no explicit representation of what the system intends to do. It is therefore difficult to ensure a good dialogue based on the intentional aspect. Other methods could be more effective. For example, if links are classified as a navigational device along with navigators and site maps, it could be a good idea to mention them whenever we speak about links, irrespective of the possible intentions of the speaker (see however Si-Saïd and Rolland (1998) for guidance based on intention tracking).

The attentional aspect deals with what people are paying attention to at a particular point in the dialogue. As shown above, this aspect may serve to direct the users' attention to related topics they may want to speak about. This aspect can also be used to restrict what can be spoken about. For example, the system could try, with sentence (c), to know everything about the publication page, now that it has been mentioned. Elicitation modules are able to deal with large specifications where problems can be detected in several places. However, users have a limited attentional capability (Miller 1956). This forces us to restrict the dialogue to what users can deal with at the time. It is relatively easy for the reasoning system to indicate which part of an utterance is the most important one. For example, in questions, the most important thing is usually the thing the question is about. This makes it possible to use an attentional method for dialogue management, even with reasoning systems which are not natural language oriented. Ignoring the attentional aspect may lead to poor dialogues (see section 6), since elicitation tools are usually not aware of human memory limitations. Users may therefore be swamped by information or they may have to spend considerable efforts understanding what the information conveyed by the tools is about. Integrating dialogue rules with the reasoning of the tools seems therefore a promissing way of improving the interaction with the tools and the way of working of their users.

Based on these considerations, we built a dialogue manager for an elicitation system, which is primarily concerned with ensuring a good attentional aspect for the interaction. The informational aspect is dealt with by the elicitation module. The intentional aspect is ignored in our system.

In our system, a good dialogue between the elicitation tool and its users is ensured by focus rules. These rules define how a dialogue can evolve (Lecœuche et al. 1998b). Three questions need to be answered when using focus rules for dialogue management:

1. On which attentional theory should we base the dialogue manager?

2. How can it be implemented?

3. Does it provide benefits over other techniques?

Although we answer these questions in the requirements engineering domain, we believe that our results have wider implications. The analysis and comparison of the focus rules carried out can be re-used in other domains. The evaluation and development method we propose can also be adapted for other purposes.

3 System Architecture

In this section, we present the general architecture of our system and we describe its main components.

The domain in which the system is used is the construction of WWW sites for research groups. The system has some notions of what a research group involves (researchers, projects, publications) and how WWW sites are designed (site, page, link)[1]. It then tries to determine how a particular research group is to be presented. The result is a specification of a WWW site, showing how the information is distributed among pages and how it is possible to navigate between them. Although we base our presentation on this domain, the system can be used in other domains.

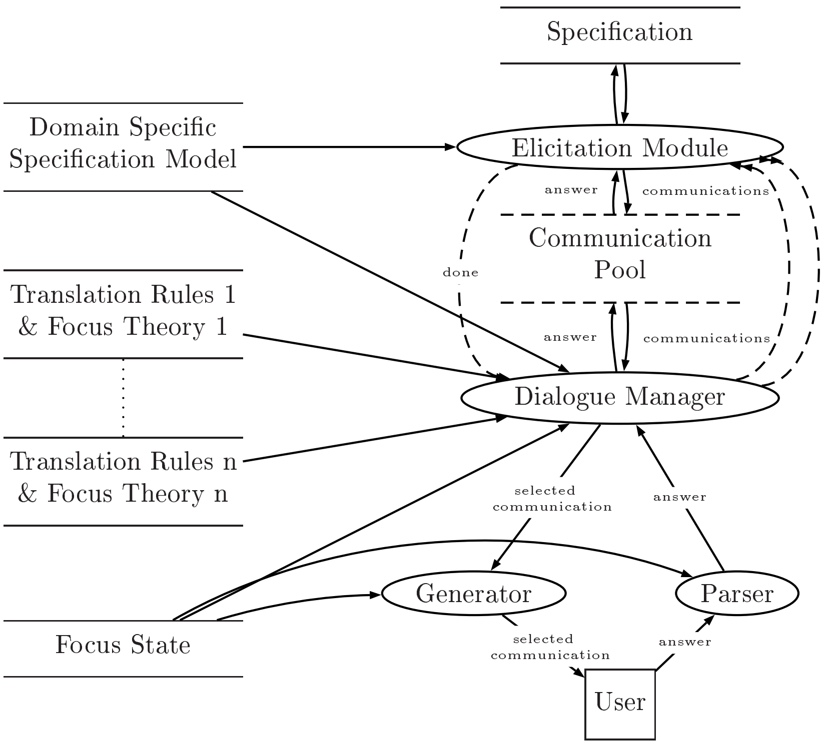

Figure 1: System architecture and transformation diagram

The general architecture of the system is shown in figure 1 (see Ward (1986) for a detailed description of the notation used). It is composed of three main parts:

- The Elicitation Module and its associated data stores (Specification and Domain Specific Specification Model). This part is in charge of eliciting the requirements and ensuring their correctness.

- The Dialogue Manager and its associated programs (Generator and Parser) and data stores (Focus Rules, Translation Rules and Focus State). It organises the dialogue between the Elicitation Module and its users.

[1]A site is a set of pages, one of them being a home page. Pages can be linked together. Several other entities exist in our system but we do not present them here.

Expression	Interpretation
$entityset(T_e)$	T_e is a set of entities of type T_e)
$relationset(T_r)$	T_r is a set of relations of type T_r
$cardmin(T_e, N)$	The minimum cardinality of T_e is N
$cardmax(T_e, N)$	The maximum cardinality of T_e is N
$attribute(T_e, A, T_a)$	Entities of type T_e have an attribute A of type T_a
$role(T_r, Role)$	Relations of type T_r have a role $Role$
$rolefiller(T_r, Role, T_e)$	Entities of type T_e can be used to fill the role $Role$ of relations of type T_r. More than one type of rolefiller can be specified
$cardmin(T_r, Role, N)$	The minimum cardinality of the role $Role$ for relation of type T_r is N
$cardmax(T_r, Role, N)$	The maximum cardinality of the role $Role$ for relation of type T_r is N
$rolecardmin(T_r, Role, N)$	The minimum cardinality of role $Role$ for relation of type T_r is N
$rolecardmax(T_r, Role, N)$	The maximum cardinality of role $Role$ for relation of type T_r is N

Table 1: ER Model notation (model level)

- The Communication Pool serves as a means of communication between the Elicitation Module and the Dialogue Manager. By enabling communications to be buffered for some time before being output, the Communication Pool plays a major role in allowing the Dialogue Manager to organise the dialogue.

We now describe the main components of the system in more detail. Its working will be explained fully in section 5.

3.1 Data Stores

The general system architecture is composed of the following data stores:

Domain Specific Specification Model This contains the domain specific knowledge of the system about what can be elicited. This model is used during the elicitation process to direct the system. It is also used for correctness and completeness checking.

In our system, this model is based on an Entity-Relationship (ER) model (Chen 1976) augmented with two additional types of constraints. The notation used to describe the model is presented in table 1. The two additional types of constraints are:

- Cardinality restrictions on types. This enables us to define the minimum and maximum numbers of entities of a particular type.
- Cardinality restrictions on relationship roles. A role is the function played by entities in a relationship. For example, in figure 2 presenting part of the model constraining the creation of a research group site, the relation

entityset(research group)	entityset(publication set)
entityset(researcher set)	entityset(site)
entityset(home page)	entityset(page)
relationset(present)	relationset(involve)
relationset(describe)	relationset(link)
cardmin(research group, 1)	cardmax(research group, 1)
cardmin(publication set, 1)	cardmax(publication set, 1)
cardmin(researcher set, 1)	cardmax(researcher set, 1)
cardmin(site, 0)	cardmax(site, ∞)
cardmin(home page, 0)	cardmax(home page, ∞)
cardmin(page, 0)	cardmax(page, ∞)
role(present, presented by)	role(present, presenting)
role(involve, involved in)	role(involve, involving)
role(describe, described by)	role(describe, describing)
role(link, linked from)	role(link, linked to)
rolefiller(present, presented by, site)	
rolefiller(present, presented by, page)	
rolefiller(present, presenting, research group)	
rolefiller(present, presenting, researcher set)	
rolefiller(present, presenting, publication set)	
rolefiller(involve, involved in, research group)	
rolefiller(involve, involving, publication set)	
rolefiller(involve, involving, researcher set)	
rolefiller(describe, described by, home page)	
rolefiller(describe, describing, site)	
rolefiller(link, linked from, home page)	
rolefiller(link, linked from, page)	
rolefiller(link, linked to, home page)	
rolefiller(link, linked to, page)	
cardmin(present, presented by, 0)	cardmax(present, presented by, 1)
cardmin(present, presenting, 0)	cardmax(present, presenting, 1)
cardmin(involve, involved in, 1)	cardmax(involve, involved in, 1)
cardmin(involve, involving, 1)	cardmax(involve, involving, 1)
cardmin(describe, described by, 0)	cardmax(describe, described by, 1)
cardmin(describe, describing, 1)	cardmax(describe, describing, 1)
cardmin(link, linked from, 0)	cardmax(link, linked from, ∞)
cardmin(link, linked to, 0)	cardmax(link, linked to, ∞)
rolecardmin(present,presenting, 1)	
attribute(home page, title, text)	
attribute(page, title, text)	

Table 2: Example of ER model in textual notation

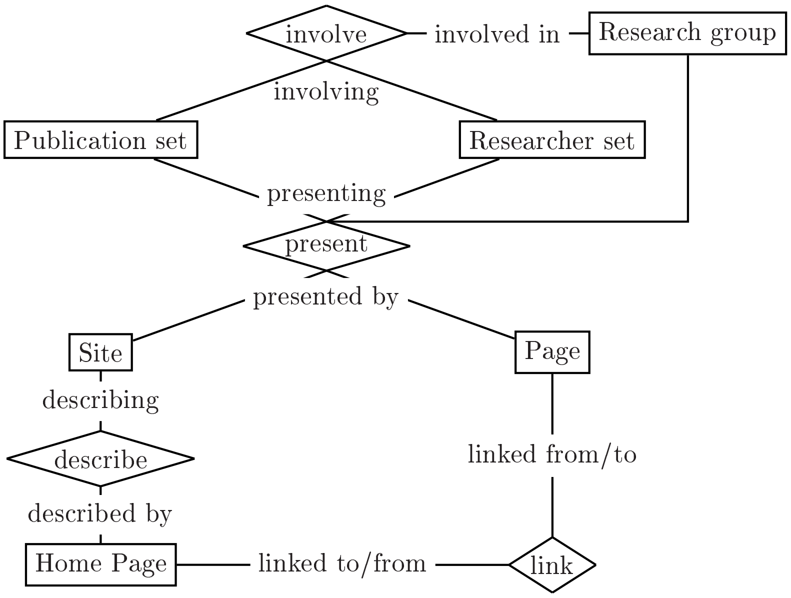

Figure 2: Example of ER model in graphical notation

`involve` has two roles, namely, `involved in` and `involving`. The latter can be filled by either `publication set` entities or `researcher set` entities. When a role can be filled by entities of several different types, the minimum and maximum numbers of rolefillers for the role can be specified. This is especially useful when roles can filled by either of two entities but not both (in this case the maximum cardinality for the role is 1). In all cases:

$$rolecardmin(Role, N) \geq \sum_{T_e} 1._{rolefiller(T_r, Role, T_e)} \times cardmin(T_r, Role, N)$$

$$rolecardmax(Role, N) \leq \sum_{T_e} 1._{rolefiller(T_r, Role, T_e)} \times cardmax(T_r, Role, N)$$

where $1._{p(x_1,\ldots,x_n)} = 1$ if $p(x_1, \ldots, x_n)$ is true and 0 otherwise. These equations mean that the role cardinality restrictions should impose more stringent constraints on the model than the constraints already imposed by the relation cardinality restriction. The equalities hold if not specified otherwise.

The part of the model constraining the creation of a research group site presented in figure 2 is detailed in table 2. (The cardinality restrictions have been omitted from the figure but appear in the table.) This example shows the use of a role cardinality restrictions to constrain sites and pages to present at least one entity, which can either be a research group, a researcher set or a publication set.

Expression	Interpretation
$entity(E, T_e)$	Entity E is of type T_e
$attributevalue(E, A, V)$	Entities E has an attribute A of value V
$relation(R, T_r)$	Relation R is of type T_r
$rolefillervalue(R, Role, E)$	Entity E fills the role *Role* of relation R

Table 3: ER Model notation (instance level)

```
entity(rg1, research group)
entity(s1, site)
relation(p1, present)
rolefillervalue(p1, presented, rg1)
rolefillervalue(p1, presented by, s1)
```

Table 4: Specification example

Specification This represents the requirements that have been elicited. In our system, the requirements are represented in an ER model. The notation used to describe the model is presented in table 3. For example, the fact that a research group is presented by a site would be represented as shown in table 4. The specification can also contain some "hypothetical" entities and relations that the elicitation system assumes will be created but have yet to be asserted by the user. These entities and relations are marked by a question mark (see section 5).

An ER model is not always the best nor the most complete way of describing requirements. We chose an ER model for its simplicity and because it is well suited for describing static domains such as the one dealt with here (Davis 1988, 1993; Wieringa 1996).

Communication Pool This contains the communications to be output by the system and the answers provided by the users.

In our system, the elicitation module and the dialogue manager interact by means of a predefined set of communication types. The elicitation module puts communications of these types in the communication pool. Each communication has one *main subject* which is either a thing to be presented or to be asked. Each communication may also have some other *subjects* which provide some context for the communication. The notation used is `communication_type(mainsubject | other subjects)`. Communications can be of two types:

- Presentation communications are used to inform users of an action made by the elicitation module, such as the creation of an entity or a relation. They can also indicate dialogue actions such as the change to a new topic. For example, `presentation(rg1)` is a communication presenting research group `rg1`.

- Question communications are used to prompt users for information. Questions serve to fill in the specification. For example, `question(?s1 | p1,`

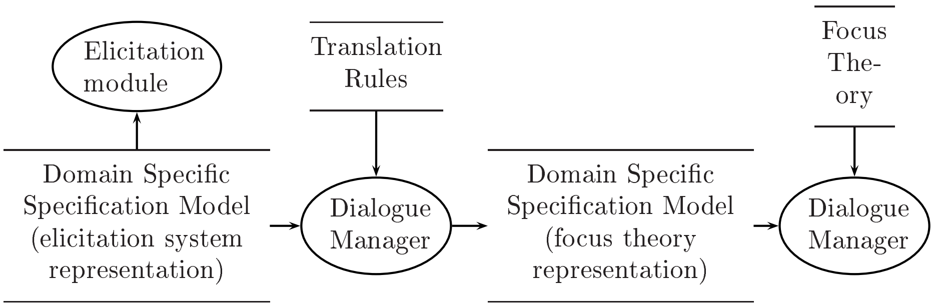

Figure 3: Translation rules

`presented by`) is a question about the identity of `?s1` with the context information, given after the vertical bar, that `?s1` is the rolefiller of role `presented by` of relation `p1`.

The communication pool also contains any users' answers to the question communications. The dialogue manager puts them in the communication pool when they become available. They are represented as `answer(communication, answer)`. The answer is either `yes` or `no` when answering yes/no questions, `new` or the identifier of an entity when answering questions about the identity of an element, or free text when giving the value of an attribute.

Translation Rules & Focus Theory The focus theories are used to rank the communications available in the communication pool in order to ensure a coherent dialogue. The theories take into account the current focus state of the dialogue to perform this task. Some focus theories need to access domain knowledge stored in the domain specific specification model. The translation rules bridge the gap between the representation used for the domain specific specification model and the representation used by the focus rules (see section 4.1.2). This is shown graphically in figure 3. Some focus theories do not use the domain specific specification model. These theories do not need translation rules.

Once a communication has been selected, the focus theories are used again by the dialogue manager to update the focus state.

The theories we have been using in our system are presented in-depth in section 4.

Focus State This data store is used to record the current focus state. It is used when applying the focus theories (see section 5).

3.2 Programs

The data stores are used by the following programs:

Elicitation Module Depending on the current state of the specification and on the domain specific specification model, the elicitation module generates communications and interprets answers contributed by the dialogue manager. Depending on these answers, it modifies the specification. It can also remove communications from the communication pool. This gives much flexibility to the elicitation system in its communication decisions, but makes the task of the dialogue manager more complex.

In our system, the elicitation system outputs all the communications it can and makes all the modifications possible to the specification given the users' answers. The general algorithm driving the module is the following.

- Create all entities and relations that should exist. This depends on the minimal cardinalities specified in the domain specific specification model.
- Add presentation communications to the communication pool presenting the things that are modified in the specification. In particular, all entity and relation creations are presented as well as relations' new role fillers.
- Add question communications to the communication pool asking for missing information, such as the creation of new entities or relations.
- Interpret answers to question communications and modify the specification accordingly.
- Remove question communications from the communication pool that would lead to a violation of constraints. This depends on the maximal cardinalities specified in the domain specific specification model and on the modifications made to the specification since the communication has been placed in the communication pool.

This elicitation module is very simple. It does not have any of the advanced reasoning capabilities that full requirement systems have. However, it is sufficient to demonstrate our point. Using more complex requirement systems would help describing requirements. It would not change the need for organising dialogues to make them easy to understand.

Dialogue Manager This program organises the dialogue by selecting a communication from the communication pool. It uses focus theories to perform this task. Several focus theories can be used together. For example, one focus theory can check for the overall coherence of the dialogue (we will call such a theory a "global focus theory") while another focus theory checks for sentence-to-sentence coherence (we will call such a theory a "local focus theory"). The dialogue manager then gives the selected communication to the Natural Language Interface (NLI) composed of a natural language generator and a natural language parser.

Our dialogue manager differs from other dialogue managers using focus rules (Huang 1994; McKeown 1985; Reed and Long 1997a) by the fact that it cannot plan reliably ahead in the text since the communication pool may be modified at any time by the elicitation module. If the dialogue manager was planning the output by relying on having several communications in the communication pool, there would always be a risk, for example, that a communication which was part of the output plan be removed.

In our system, the dialogue manager has access to different focus strategies that can be "plugged-in" to the system. The dialogue manager uses them in a predefined order. (This order can however be changed at run-time by the system). Theory 1 gives a first partial ranking which is then refined by theory 2 and so on until the last theory N. The highest ranked communication is then selected and output. Using this method we are able to generate texts based only on the local context, without planning (Sibun 1992), by using local focus theories, but we are also able to ensure global coherence by adding global focus theories (see section 4). Therefore, we can adapt the level of text structure depending on the need of the dialogue. This, of course, depends on the communications available in the communication pool and therefore to a certain extend to the elicitation module (see section 5).

In the case where focus theories are unable to distinguish between competing communications, presentation communications are preferred over question communications. This is because it seems better to provide information before asking questions. If communications are still competing, one is selected at random.

Generator The role of the generator is to output the communications in natural language.

In our system, the output is based on templates which can be adapted depending on the context (Reiter 1995). For example, the generator makes use of the focus theories results (if available) to make decisions about pronominalisation, i.e., the use of pronouns such as "it", and clue words selection, i.e., the use of words such as "now" and "anyway".

Parser The role of the parser is to interpret the answers of the users based on a controlled language. It transforms them into a form that can be used by the dialogue manager and elicitation module.

In our system, the parsing is based on a set of templates. The templates are divided into three groups:

- Question answering templates enable users to answer questions asked by the system. An example of a sentence that a user could enter is: "Research group `rg1` is presented by a new site" as an answer to "Which site presents research group `rg1`?".

- Automatic question answering templates enable users to set the NLI so that it automatically answers questions without outputing them. An example of a sentence that a user could enter is: "Every new page is linked to the homepage". The NLI won't ask about linking pages to the homepage anymore but will automatically answer `yes`.

- Dialogue directing templates enable users to shift the focus of the dialogue to a new topic. (This capability depends on the focus theories used.) An example of a sentence that a user could enter is: "I want to speak about the research group".

The parser also performs some anaphora resolution, i.e., finds the nouns pronouns refer to, based on the available focus information.

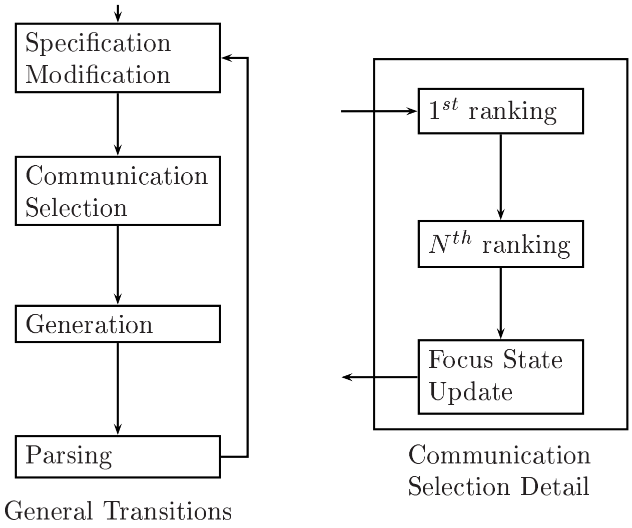

Figure 4: System state diagram

3.3 Activation

The various elements presented above are activated in a cycle shown in figure 4 and described below:

1. The elicitation module is activated by the dialogue manager. It then modifies the specification and modifies the communication pool if needed. For example, at the start of the elicitation process, the system creates an entity representing a research group and puts a communication presenting it in the communication pool. The elicitation module then sends a signal to the dialogue manager indicating the end of the modifications. It is then de-activated.

2. The dialogue manager selects a communication. This is done by using the focus theories to rank the communications. The details of this process are presented in figure 4. In the case of the communication created above, it would be selected since this is, for now, the only one in the communication pool. Once a communication has been selected, the focus state is updated.

3. The generator outputs the communication. The generator would output here something like “Let’s speak about the research group”.

4. The parser interprets the user’s answer. In this case the user could simply acknowledge the communication or try to redirect the dialogue by providing new information.

The dialogue manager plays a pivotal role in this cycle by activating the other elements when needed, as shown in figure 1.

4 Formalising focus theories

The attentional aspect of dialogues is the subject of numerous theories, called focus theories. We made a careful survey of many focus theories (Lecœuche et al. 1998a). In particular, we distinguished between global focus theories which organise dialogues at a high level, and local focus theories which organise dialogues on a local basis. Both kinds of theories are needed to ensure a coherent dialogue. Unfortunately, most of these theories are not stated clearly enough to be easily implemented. In this section, we present the formalisation of two theories. We based these formalisations on two well-known focus theories. We chose these theories because they were already quite clearly stated, and mostly domain-independent. One is a global focus theory (see section 4.1) and the other is a local focus theory (see section 4.2).

An extended example of the use of these theories is given in section 5.

4.1 Global Theory

Our formalisation of a global focus theory is based on a simplification of Reichman's global focus theory (Reichman 1985). This theory is a well-established theory of global focus. It is powerful enough to deal with a lot of dialogue phenomena while remaining simple enough to be implemented. The basic idea is that the dialogue is supported by a sequence of changes to a focus space set. The focus space set contains focus spaces which represent the information in focus during part of the dialogue. Each focus space contains all the things to which participants in a dialogue are attending, at a certain point in a dialogue. If $\mathcal{S} = \{\mathcal{F}_1\}$ is the initial focus space set, only containing the initial focus space $\mathcal{F}_1$ at the beginning of the dialogue, then the set when t exchanges have been carried out in the dialogue is $\mathcal{S} = \{\mathcal{F}_1, \ldots, \mathcal{F}_n\}$ where each $\mathcal{F}_i$ contains some of the things spoken about in the dialogue. We may need to create several focus spaces ($\mathcal{F}_i$) since the dialogue may deal with more than one topic. Focus spaces have different activation levels. We use three activation levels in our formalisation:

Active This is the space to which current communications are added. This space is unique at any given point in the dialogue. For example, at the beginning of the dialogue, focus space $\mathcal{F}_1$ is the active space.

Controlling These are the spaces expected to become active again when the space they control is closed. The controlling spaces form a tree: each space is controlled by at most one other space and a space cannot control a space already controlling it or controlling any of its controlling spaces. The controlling spaces serve one of two roles: they represent the point of view we have on things that are discussed in the active focus space, or they simply represent things we have to return to after an interruption (see rule 4.6 below). These two functions are grouped together in the notion of controlling spaces because they give rise to the same dialogue structure, i.e., a return to the controlling space.

Closed These are the spaces which have been dealt with and are not expected to be returned to.

Communications cause things to be included in focus spaces. They can also cause focus space activations to be modified and/or focus spaces to be created.

Our theory is composed of seven rules. The focus rules define the possible ways the dialogue may develop and the constraints to satisfy for choosing a particular way. Associated with an ordering, they represent what we are expecting to say next in the dialogue. Four rules create new spaces and two change the activation levels of existing spaces. The four creating moves are differentiated along two criteria: (1) moves that keep/do not keep the current focused entities in focus, (2) moves that are expected/not expected to shift back to the current focus space once the newly created space is closed. The moves are based on four relations between the things that can be mentioned in the dialogue. Unlike ER relations, these relations are directional: for example, if a thing is in a specialisation relation with another there is not always a specialisation relation from that other thing to the first. The four relations are:

Direct relation There is a direct relation from one thing to another if the other is closely related to it and can be mentioned in the same focus space. (This means in particular that a thing is in direct relation with itself.) Conversely, if two things belong to the same focus space, they are considered to be in a direct relation.

Specialisation relation There is a specialisation relation from one thing to another if the other is more specific than it. In that case, the more specific thing can be discussed in the perspective of the more generic one.

Generalisation relation There is a generalisation relation from one thing to another if there is a specialisation relation in the other direction between the two things and no direct or specialisation relation links the first thing to the other.

Simple relation There is a simple relation from one thing to another if it is related to the other and no direct, specialisation or generalisation relation links it to the other.

The exact nature of these relations depends on the application domain. The issue of translating between domain relations and these relations is discussed in section 4.1.2.

Because focus rules only indicate what changes in the focus spaces (most things stay the same), we are faced with a simple instance of the frame problem, i.e., how to indicate what changes and what does not (Kowalski 1979, ch. 6). We tackle this problem by taking into account the time for which a relation holds and adding information which indicates the end of this time period rather than deleting the relation itself (Kowalski and Sergot 1986). In particular, we consider the time in the dialogue as being the sequence of communications and associated focus moves that have been performed since the beginning of the dialogue. For example, at the start of the dialogue, the time is the empty sequence $\langle\rangle$ and after communications $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ have been output with their associated moves m_1, m_2 and m_3 respectively, the time is the sequence $\langle m_1(\mathcal{C}_1), m_2(\mathcal{C}_2), m_3(\mathcal{C}_3)\rangle$. In other words, if the current time is t and the next communication output is $\mathcal{C}_u$ and the output is associated with move m, then the new time is $t \cdot \langle m(\mathcal{C}_u)\rangle$ where $\cdot$ is the sequence concatenation operator. From a possible worlds perspective, the accessibility relation between worlds

Expression	Interpretation
$\mathcal{F}_i$	A focus space
$active(\mathcal{F}_i)$	Focus space $\mathcal{F}_i$ is active
$controlling(\mathcal{F}_i, \mathcal{F}_j)$	Focus space $\mathcal{F}_i$ is controlling focus space $\mathcal{F}_j$
$closed(\mathcal{F}_i)$	Focus space $\mathcal{F}_i$ is closed
$\mathcal{C}$	$\mathcal{C}$ is a communication, i.e., a message to be output. It makes the dialogue evolve from time t to $t+1$.
$subject(\mathcal{C}, X)$	X is a subject of communication $\mathcal{C}$
$mainsubject(\mathcal{C}, X)$	X is the main subject of communication $\mathcal{C}$. The communication main subject is a communication subject
$dir(D, X_1, X_2)$	X_1 is in direct relation D with X_2
$spe(S, X_1, X_2)$	X_1 is in specialisation relation S with X_2
$gen(G, X_1, X_2)$	X_1 is in generalisation relation G with X_2
$sim(R, X_1, X_2)$	X_1 is related to X_2 by simple relation R

Table 5: Notation used to formalise the global focus rules

corresponding to this notion of time is defined as $W_1\ R\ W_2 \Leftrightarrow time(W_1, t_1)\ \wedge\ time(W_2, t_2)\ \wedge\ t_2 = t_1 \cdot t$ where the *time* predicate associates a world with its time and t is a non empty sequence of moves. This relation is anti-reflexive ($\forall W.\neg(W\ R\ W)$), antisymmetric ($\forall W_1, W_2.W_1\ R\ W_2 \rightarrow \neg(W_2\ R\ W_1)$), transitive ($\forall W_1, W_2, W_3.(W_1\ R\ W_2\ \wedge\ W_2\ R\ W_3) \rightarrow W_1\ R\ W_3$) and linear on the left ($\forall W_1, W_2, W_3.(W_1\ R\ W_3 \wedge W_2\ R\ W_3) \rightarrow (W_1 = W_2 \vee W_1\ R\ W_2 \vee W_2\ R\ W_1)$). From these properties we can deduce (Haton et al. 1991, p. 189) that each world has only one past (the dialogue history) and several possible futures (depending on which moves are available).

We define the $\succeq$ relation on time in the following way: $t_1 \succeq t_2 \Leftrightarrow \exists t_3.(t_1 = t_2 \cdot t_3)$. The relation $\succ$ is then defined as $t_1 \succ t_2 \Leftrightarrow t_1 \succeq t_2\ \wedge\ \neg(t_1 = t_2)$. Relations that begin to hold at a certain time are written $start(P, t)$ where P is the relation and t the time. Relations that stop holding at a certain time are written $end(P, t)$ where P is again the relation and t the time. In order to know if a relation holds at a certain time, we define the predicate $hold(P, t)$. The definition of this predicate is $hold(P, t) \Leftrightarrow \exists t_{start}.(t \succeq t_{start}\ \wedge\ start(P, t_{start})\ \wedge\ \neg\exists t_{end}.(t \succeq t_{end}\ \wedge\ t_{end} \succ t_{start}\ \wedge\ end(P, t_{end})))$.

Because the notation used above is quite cumbersome, we will use some shortcuts. In particular, we will write $hold(P(A_1, \ldots, A_n), t)$ as $P_t(A_1, \ldots, A_n)$. The same applies for infix operators. For example, inclusion in a set at time t is noted $\in_t$. We will also often ignore the exact nature of the time sequence, only considering how many communications have been output. The exact sequence can be re-constructed from the dialogue history so we do not lose any information in doing so. For example, the time at the start of the dialogue will be $t = 0$ and after three communications have been output $t = 3$. We will ignore the time altogether if it does not influence the truth value of the predicate.

The notation used to represent the rules used in the theory is shown in table 5. Some immediate properties of the theory are given in table 6.

In the following rules, the time variable, t, the communication variable, $\mathcal{C}$, and all variables appearing on both sides of the implications are universally quantified at the outer most level. All other variables are assumed to be exis-

$\forall t.(t \geq 0 \rightarrow \exists \mathcal{F}_i.(active_t(\mathcal{F}_i) \wedge \forall \mathcal{F}_j.(active_t(\mathcal{F}_j) \rightarrow \mathcal{F}_j = \mathcal{F}_i)))$	At any point in the dialogue there is one and only one active focus space
$\forall t, X_1.\exists D.dir_t(D, X_1, X_1)$	Any element is in direct relation with itself
$\forall t, X_1, X_2.((X_1 \in_t \mathcal{F}_i \wedge X_2 \in_t \mathcal{F}_i) \rightarrow \exists D.dir_t(D, X_1, X_2))$	If two elements are in the same focus space, they are in some direct relation
$\forall t, X_1, X_2.(\exists S.spe_t(S, X_1, X_2) \rightarrow \exists G.gen_t(G, X_2, X_1))$	If there is a specialisation relation between X_1 and X_2, there is a generalisation relation between X_2 and X_1

Table 6: Immediate properties of global focus theory

tentially quantified unless explicitly universally quantified. When a new space is created, it is assigned number $n+1$ where n is the number of existing spaces. n can be deduced from the dialogue time since it is equal to the number of moves in the dialogue history opening new focus spaces. The value of $n+1$ is computed by the function $new\ index(t)$ where t is the dialogue time. The interpretation of rule 4.1 (no change) for example is then the following: "If the current active focus space is $\mathcal{F}_i$ and if we can select a communication $\mathcal{C}$ such that its main subject is in direct relation with an element of $\mathcal{F}_i$, then we can perform a no change focus move. This move has the effect of putting all the subjects of $\mathcal{C}$ in space $\mathcal{F}_i$."

Rule 4.1 (No change) *The focus space does not change and new information is added to it. This rule is used to speak about things closely related to the things in the current active focus space.*

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ dir_t(D, X_1, X_2) \end{array}\right) \rightarrow \forall X.(subject(\mathcal{C}, X) \rightarrow start(X \in \mathcal{F}_i, t'))$$

where t' stands for $t \cdot \langle no\ change(\mathcal{C})\rangle$.

Rule 4.2 (Resetting) *A new focus space is created. This rule is used to speak about more abstract things than the ones in the current active focus. It may therefore serve to give background information on the things in the current active focus.*

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ gen_t(G, X_1, X_2) \end{array}\right) \rightarrow \left(\begin{array}{l} end(active(\mathcal{F}_i, t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \quad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \end{array}\right)$$

where t' stands for $t \cdot \langle resetting(\mathcal{C})\rangle)$.

Rule 4.3 (Additive) *A new focus space is created. It is controlled by the current active space. Entities in the current active space are copied to the new space. This rule is used to speak about things that are more precise than the* *ones in the current active focus space. The new things are discussed from the perspective of the current active focus space.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ spe_t(S, X_1, X_2) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ start(controlling(\mathcal{F}_i, \mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(X \in_t \mathcal{F}_i \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \end{pmatrix}$$

where t' stands for $t \cdot \langle additive(\mathcal{C})\rangle$.

Rule 4.4 (Generating) *A new focus space is created. Entities in the current active space are copied to the new space. Any controlling relation is passed from the current active space to the newly created space since we may not expect to come back to the current active space but we still expect to come back to its potential controlling space. This rule is used to speak about things related to the things in the current active focus space but not closely associated with them.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ sim_t(R, X_1, X_2) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(X \in_t \mathcal{F}_i \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \\ \wedge\ controlling_t(\mathcal{F}_j, \mathcal{F}_i) \\ \rightarrow start(controlling(\mathcal{F}_j, \mathcal{F}_{new\ index(t')}), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle generating(\mathcal{C})\rangle$.

Rule 4.5 (Pop) *A controlling space becomes active again. This rule is used to come back to a space that was expected to be reactivated. This rule is always used in conjunction with one of the other rules.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ controlling_t(\mathcal{F}_j, \mathcal{F}_i) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_j), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle pop(\mathcal{C})\rangle$.

Rule 4.6 (Digressing) *A new focus space is created. It is controlled by the current active space. This rule is used to change the focus of the dialogue for a period of time after which the dialogue will resume where it was interrupted.*

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ mainsubject(\mathcal{C}, X_1) \wedge \\ \forall R.(\neg\exists X.(X \in_t \mathcal{F}_i \wedge \\ (dir_t(R, X, X_1) \vee spe_t(R, X, X_1) \\ \vee gen_t(R, X, X_1) \vee sim_t(R, X, X_1)))) \wedge \\ \forall j.(X_1 \notin_t \mathcal{F}_j) \end{array}\right) \rightarrow$$

$$\left(\begin{array}{l} end(active(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ start(controlling(\mathcal{F}_i, \mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \end{array}\right)$$

where t' stands for $t \cdot \langle digressing(\mathcal{C})\rangle$.

Rule 4.7 (Reopening) *An old space becomes active again. This rule is used to come back to a topic that was considered dealt with. There are two ways a topic may be re-introduced in a dialogue: (1) we realise we forgot to say something about it and come back to it, (2) the topic is discussed from another perspective. The reopening move only addresses the first type of re-introduction. For the second type a new focus space would be created since the dialogue context (i.e., the controlling space) has changed.*

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ closed_t(\mathcal{F}_j) \wedge \\ mainsubject(\mathcal{C}, X_1) \wedge \\ X_1 \in_t \mathcal{F}_j \wedge \\ \forall R.(\neg\exists X.(X \in_t \mathcal{F}_i \wedge \\ (dir_t(R, X, X_1) \vee spe_t(R, X, X_1) \\ \vee gen_t(R, X, X_1) \vee sim_t(R, X, X_1)))) \end{array}\right) \rightarrow \left(\begin{array}{l} end(active(\mathcal{F}_i), t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ end(closed(\mathcal{F}_j), t') \\ \wedge\ start(active(\mathcal{F}_j), t') \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \quad \rightarrow start(X \in \mathcal{F}_j, t')) \end{array}\right)$$

where t' stands for $t \cdot \langle reopening(\mathcal{C})\rangle$.

The dialogue starting state for this theory is $active_0(\mathcal{F}_1)$ with $\mathcal{F}_1 =_0 \{\}$.

Although the main characteristics of Reichman's approach are conserved, the theory presented above is a simplified version of the original approach. In particular, fewer types of focus spaces and fewer types of relations between focus spaces are used. Also, each focus space contains less information than in Reichman's theory. Compared to other approaches, our focus spaces contain also less information than in Hitzeman and Poesio's approach (Hitzeman and Poesio 1998) which is closed to Reichman's. On the other hand, this theory is richer than Grosz and Sidner's approach (Grosz and Sidner 1986) since we distinguish between normal evolution of the dialogue and digressions. We also keep track of closed spaces. This balance between simplicity and complexity seems adequate for our purposes.

Because the theory is formalised, it is possible to prove that some properties hold. For example, we can prove that, at any point in the dialogue, there is one and only one active focus space (Lecœuche et al. 1999).

4.1.1 Ordering of the rules

Several rules may have their preconditions satisfied at the same time. We prefer then to apply the rule that maintains the focus if possible, or minimises its movement. Changing focus, i.e., changing the entities to which participants are paying attention, does require some cognitive processing. Therefore, minimising focus movement reduces the cognitive overhead needed to understand the dialogue. We minimise the focus movements by presenting general concepts before specialised ones and by avoiding references to unrelated concepts. Rules are therefore applied in the following preference order:

> no change > resetting > additive > generating > pop > digressing > reopening.

This means that we first try to find a communication to be output which would allow a "no change" move. Then, if such a communication does not exist, we would try to find a "resetting" communication and so on until a communication is found.

There is an exception to this ordering. If the main subject of a possible communication to output is already a member of a controlling space then a pop is the preferred move. This avoids reintroducing concepts that we are expecting to return to later in the dialogue. An example of the use of this theory is given in section 5.

4.1.2 Translation rules

Organising a dialogue based on attention requires a lot of domain knowledge, in order to know how things mentioned in the dialogue relate to each other. Therefore, careful engineering is needed to build these kinds of natural language applications. We have tried to limit the engineering difficulties by providing a way of re-using the existing domain knowledge as far as possible. This is done by using "translation" rules which enable us to re-use part of the domain knowledge already used by the reasoning module. Because the focus rules we presented in section 4.1 are based on generic concepts such as direct, specialisation, generalisation and simple relations which are not necessarily used by the elicitation module, we need to bridge the gap between the representation used by the elicitation module and the representation used by the focus rules (Kittredge et al. 1991). This is done by the translation rules. These rules also allow the use of this focus theory in different domains.

We have created a set of translation rules to map the ER representation used by the elicitation system (see section 3) to the generic concepts used by our formalised focus theory. Basing the translation rules on the structure of the ER model has the advantage of not being dependent on the semantics of the relations which is a major problem for some theories (Polanyi 1985; Schank 1977). We are still dependent on the semantics of the ER model. However, this dependency is at a higher level than domain specific ones. It is therefore usually easier to adapt the system to new domains. This reduces the amount of engineering work required to use our approach in various domains. For example, using the rules presented here we have been able to use our system in two other domains than the WWW site design: the elicitation of a simple library model and the elicitation of a generic ER model, which could then be used to elicit a particular specification. Applying our approach to a framework for which translation rules have not been created would require much more work. Moreover, in some domains, the generator and the parser may need to be adapted to reflect domain linguistic idiosyncrasies. In that case, our approach would be worthwhile only if its use makes up for the investment needed up-front. This could be the case in domains where the requirements are likely to change often. Basing the translation rules on the structure of the ER model can also be a drawback for domains where the structure of the model is less well related to the structure of the dialogue than in our application domain. For example, attributes are treated as independent in our system. This could be a problem if attributes can be grouped into clusters. For example, some attributes could deal with colour of objects (background colour, border, colour, etc.) while others could deal with their size (length, height, etc.). Mixing these attributes would make the text less coherent than if they were discussed in two clusters. In this case, translation rules would have to be more domain-dependent to classify the attributes in their clusters. Reaching a balance between domain dependence and translation power is a difficult task.

The main idea behind the rules presented in this section is to allow relations and entities that are related to the current focus and cannot be accessed easily otherwise to be mentioned immediately. (Their actual selection will depend on the current focus state and the focus rules.) The translation rules are presented below. In the following rules, all variables are assumed to be universally quantified unless explicitly existentially quantified.

Rule D1 If two entities are in the same space, any relation linking them is in a direct relation with them.

$$\left(\begin{array}{r} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ E_1 \in \mathcal{F}_{i,t} \wedge \\ E_2 \in \mathcal{F}_{i,t} \end{array}\right) \rightarrow \left(\begin{array}{l} \exists D.dir_t(D, E_1, R) \\ \wedge\ \exists D'.dir_t(D', E_2, R) \end{array}\right)$$

Rule D2 If an entity can be linked by a relation to another one, is optional, i.e., it may not exist at all, and is dependent on the other one, i.e., it cannot exist without it, then it is in a direct relation with the relation.

$$\left(\begin{array}{r} relation_t(R, T_r) \wedge \\ role(T_r, Role_1) \wedge \\ role(T_r, Role_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ rolecardmin(Role_1, 1) \wedge \\ cardmax(T_r, Role_1, 1) \wedge \\ cardmin(T_r, Role_2, 0) \wedge \\ cardmax(T_r, Role_2, 1) \wedge \\ rolefillervalue_t(R, Role_1, E_1) \end{array}\right) \rightarrow \exists D.dir_t(D, E_1, R)$$

For example, in the specification model presented in table 2, a home page is an optional entity dependent on the existence of a site. A site is itself an optional entity dependent on the existence of a research group, a publication set or a researcher set.

Rule D3 Relation role-fillers are in direct relations with their associated relation.

$$rolefillervalue_t(R, Role, E) \rightarrow \exists D.dir_t(D, R, E)$$

There are three rules for creating specialisation relations:

Rule S1 If an entity can be linked to several others playing the same role, then there is a specialisation relation between this entity and the relations linking it to the others. This is because we consider this entity more central than the several other entities that can be related to it.

$$\left(\begin{array}{r} relation_t(R, T_r) \wedge \\ role(T_r, Role_1) \wedge \\ role(T_r, Role_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ rolecardmax(Role_2, C) \wedge \\ C > 2 \wedge \\ rolefillervalue_t(R, Role_1, E_1) \end{array}\right) \rightarrow \exists S.spe_t(S, E_1, R)$$

For example, in the specification model presented in table 2, the research group involves the researcher set and the publication set which fill the same role `involving`. The research group is therefore in a specialisation relation with the relations linking it to the researcher set and the publication set.

Rule S2 If an entity can be linked by a relation to another one, is optional but not dependent on the other one, it is in a specialisation relation with the relation. Since the entity can exist on its own, it is not included in the other entity focus space. However, since it is optional, discussing it in the light of the other entity seems a good way of introducing it.

$$\left(\begin{array}{r} relation_t(R, T_r) \wedge \\ role(T_r, Role_1) \wedge \\ role(T_r, Role_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ cardmin(T_r, Role_1, 0) \wedge \\ cardmax(T_r, Role_1, 1) \wedge \\ rolecardmin(T_r, Role_2, 0) \wedge \\ cardmax(T_r, Role_2, 1) \wedge \\ rolefillervalue_t(R, Role_1, E_1) \end{array}\right) \rightarrow \exists S.spe_t(S, E_1, R)$$

Rule S3 An entity is in a specialisation relation with another entity if it is linked to the other by a relation which is itself in a specialisation relation with the entity.

$$\left(\begin{array}{r} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ spe_t(S, E_1, R) \end{array}\right) \rightarrow \exists S'.spe_t(S', E_1, E_2)$$

There are two rules for creating generalisation relations:

Rule G1 Generalisation relations are the reverse of specialisation relations.

$$spe_t(S, E_1, E_2) \quad \rightarrow \exists G.gen_t(G, E_2, E_1)$$

Rule G2 Rule G1 is modified for the entity-relation case: An entity is in a generalisation relation with a relation if the other role filler is in a specialisation relation with this relation.

$$\begin{pmatrix} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ \hfill Role_1 \neq Role_2 \wedge \\ \hfill spe_t(S, E_2, R) \end{pmatrix} \rightarrow \exists G.gen_t(G, E_1, R)$$

Finally, there are two rules for creating simple relations:

Rule R1 An entity and a relation are in relation if the entity is a role filler of the relation.

$$rolefillervalue_t(R, Role, E) \rightarrow \quad \exists R.sim_t(R, E, R)$$

Rule R2 Two entities are in relation if they are linked by a relation.

$$\begin{pmatrix} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ Role_1 \neq Role_2 \end{pmatrix} \rightarrow \exists R.sim_t(R, E_1, E_2)$$

Changing the translation rules gives to the theory a more local or more global aspect. If more direct relations are generated, entities and relations will be grouped in a few large focus spaces. If more specialisation relations are generated, entities and relations will be grouped in numerous small focus spaces. In our case, the former case is preferred as local focus decisions are made by a local focus theory (see section 4.2).

Applying these rules to the ER model presented in figure 2, we obtain the relations presented in figure 5. For example, the research group is in a specialisation relation with the researcher set because the research group involves other sets, e.g, publication set, so that rolecardmax(involve,C), C >= 2 and translation rule S1 applies.

4.1.3 Surface phenomena

Focus theories not only help in organising the dialogue but also indicate how this organisation can be made explicit in the dialogue. For example, creating a new focus space can be marked in the dialogue by a cue word such as "now". Because the focus theory organises the dialogue into focus spaces, it is also possible to know when a topic is returned to as a normal part of the dialogue (as opposed to re-opening a closed focus space). In that case, the natural language generator can provide useful cues as shown in example 2 where it explicitly redirects the dialogue. (This use of cue words is also made in Cohen (1984, 1987); Reed and Long (1997b).)

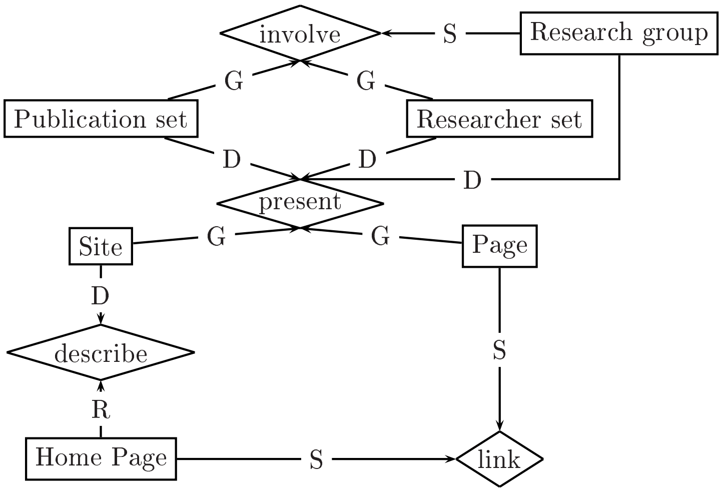

Figure 5: Direct, Specialisation, Generalisation and other relations

The figure represents the by-default direct (D), specialisation (S), generalisation (G) and other (R) relations from entities to relations (arrow direction). There is always a direct relation from relations to entities (opposite to arrow direction; translation rule D3). The relations may change as the dialogue evolves, depending on what is in focus (focus spaces general properties and translation rule D1).

Example 2

S: *Let's come back to our previous topic*
Researcher set `rs1` is involved in research group `rg1`

4.2 Local theory

Our formalisation is based on the centering theory (Grosz et al. 1995). This theory is a major local focus theory, i.e., a theory organising dialogues on a local basis. The basic idea is that each sentence introduces things into the dialogue. The next sentence then selects one of these things as the topic of the dialogue. For example, assuming I just said: "I bought a new hat at the shop round the corner yesterday", if the next sentence is: "I am going to a wedding" then the dialogue is about me but if the next sentence is: "It is nice, isn't it" then the dialogue is about the hat.

Each sentence in the dialogue is therefore associated with some "forward-looking centres", i.e., the things introduced by the sentence, and to a "backward-looking centre", i.e., the topic selected by the sentence from the previous sentence's forward-looking centres. Forward-looking centres are ranked. The most highly ranked centre is called the preferred centre and is supposed usually to become the backward-looking centre of the next sentence, although this is not

Expression	Interpretation
$Cb(X)$	X is the backward-looking centre
$Cf(X_1, \ldots, X_n)$	$X_1, \cdots, X_n$ are the forward-looking centres
$Cp(X)$	X is the preferred centre
$\mathcal{C}$	$\mathcal{C}$ is a communication, i.e., a message to be output. It makes the dialogue evolve from time t to $t+1$.
$Cb(\mathcal{C}, X)$	X would be the new backward-looking centre if communication $\mathcal{C}$ was output
$Cf(\mathcal{C}, X_1, \ldots, X_n)$	$X_1, \cdots, X_n$ would be the new forward-looking centres if communication $\mathcal{C}$ was output
$Cp(\mathcal{C}, X)$	X would be the new preferred centre if communication $\mathcal{C}$ was output
$subject(\mathcal{C}, X)$	X is a subject of communication $\mathcal{C}$
$mainsubject(\mathcal{C}, X)$	X is the main subject of communication $\mathcal{C}$. The communication main subject is a communication subject

Table 7: Notation used to formalise the local focus rules

necessary. The ranking of the forward-looking centres depends on the importance of the things introduced.

When a new communication is output, the new backward-looking centre is the most highly ranked element of the current forward-looking centres realised in the new sentence. The new forward-looking centres are the things introduced by the new sentence.

The theory is composed of five rules. These rules and the ordering associated with them constrain how the dialogue may evolve.

The notation used to represent the rules is shown in table 7. The rules are again only indicating what changes in the dialogue state. We use the same time framework as in the global focus theory formalisation (see section 4.1) to deal with the frame problem. Note that if both theories are used at the same time, the dialogue time becomes a sequence of tuples where each tuple is composed of a global focus move and a local focus move. This can be generalised for more than two focus theories. Some immediate properties of the local focus theory are given in table 8. In the following rules, the time variable, t, the communication variable, $\mathcal{C}$, and all variables appearing on both sides of the implications are universally quantified at the outer most level. All other variables are assumed to be existentially quantified unless explicitly universally quantified. The interpretation of rule 4.8 (continuation) for example is then the following: "If the current backward-looking centre is B and if we can select a communication $\mathcal{C}$ such that its backward-looking centre with respect to the current dialogue state and its preferred centre are both equal to B then we can perform a continuation focus move. This move has the effect of changing the current forward-looking centres to those of $\mathcal{C}$ and the preferred centre to that of $\mathcal{C}$."

Rule 4.8 (Continuation) *The new backward-looking centre is the same as for the previous sentence and is also the new preferred centre. This rule is used to continue to speak about the same thing.*

$\forall t, X, X_1, \ldots, X_n.(Cp_t(X) \wedge Cf_t(X_1, \ldots, X_n) \rightarrow X \in \{X_1, \ldots, X_n\})$	The preferred centre is always one of the forward-looking centres

Table 8: Immediate properties of local focus theory

$$\left(\begin{array}{r} Cb_t(B) \wedge \\ Cp_t(P) \wedge \\ Cf_t(F) \wedge \\ Cb_t(\mathcal{C}, NewB) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \wedge \\ B = NewB = NewP \end{array}\right) \rightarrow \left(\begin{array}{l} end(Cp(P), t') \\ \wedge\, start(Cp(NewP), t') \\ \wedge\, end(Cf(F), t') \\ \wedge\, start(Cf(NewF), t') \end{array}\right)$$

where t' stands for $t \cdot \langle continuation(\mathcal{C})\rangle$.

Rule 4.9 (Retaining) *The new backward-looking centre is the same as for the previous sentence but it is different from the new preferred centre. This rule is used to introduce a new topic in the dialogue with respect to the current one.*

$$\left(\begin{array}{r} Cb_t(B) \wedge \\ Cp_t(P) \wedge \\ Cf_t(F) \wedge \\ Cb_t(\mathcal{C}, NewB) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \wedge \\ B = NewB \wedge \\ \neg NewB = NewP \end{array}\right) \rightarrow \left(\begin{array}{l} end(Cp(P), t') \\ \wedge\, start(Cp(NewP), t') \\ \wedge\, end(Cf(F), t') \\ \wedge\, start(Cf(NewF), t') \end{array}\right)$$

where t' stands for $t \cdot \langle retaining(\mathcal{C})\rangle$.

Rule 4.10 (Smooth-shift) *The new backward-looking centre is not the same as for the previous sentence but is also the same as the new preferred centre. This rule is used to shift the focus of the dialogue to a new topic which is supposed to be discussed for some time.*

$$\left(\begin{array}{r} Cb_t(B) \wedge \\ Cp_t(P) \wedge \\ Cf_t(F) \wedge \\ Cb_t(\mathcal{C}, NewB) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \wedge \\ \neg B = NewB \wedge \\ NewB = NewP \end{array}\right) \rightarrow \left(\begin{array}{l} end(Cb(B), t') \\ \wedge\, start(Cb(NewB), t') \\ \wedge\, end(Cp(P), t') \\ \wedge\, start(Cp(NewP), t') \\ \wedge\, end(Cf(F), t') \\ \wedge\, start(Cf(NewF), t') \end{array}\right)$$

where t' stands for $t \cdot \langle smooth\,shift(\mathcal{C})\rangle$.

Rule 4.11 (Rough-shift) *The new backward-looking centre is not the same as for the previous sentence and is different from the new preferred centre. This* *rule is used to shift the focus of the dialogue to some new topic.*

$$\begin{pmatrix} Cb_t(B) \wedge \\ Cp_t(P) \wedge \\ Cf_t(F) \wedge \\ Cb_t(\mathcal{C}, NewB) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \wedge \\ \neg B = NewB \wedge \\ \neg NewB = NewP \end{pmatrix} \rightarrow \begin{pmatrix} end(Cb(B), t') \\ \wedge\, start(Cb(NewB), t') \\ \wedge\, end(Cp(P), t') \\ \wedge\, start(Cp(NewP), t') \\ \wedge\, end(Cf(F), t') \\ \wedge\, start(Cf(NewF), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle rough\, shift(\mathcal{C})\rangle$.

Rule 4.12 (No centre) *The backward-looking centre cannot be computed. It is considered by-default to be the same as the preferred centre. This rules is used to determine the by-default focus of the dialogue when none can be computed otherwise.*

$$\begin{pmatrix} Cb_t(B) \\ Cp_t(P) \\ Cf_t(F) \\ Cb_t(\mathcal{C}, \emptyset) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \end{pmatrix} \rightarrow \begin{pmatrix} end(Cb(B), t') \\ \wedge\, start(Cb(NewP), t') \\ \wedge\, end(Cp(P), t') \\ \wedge\, start(Cp(NewP), t') \\ \wedge\, end(Cf(F), t') \\ \wedge\, start(Cf(NewF), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle no\, centre(\mathcal{C})\rangle$.

The dialogue starting state for this theory is $Cb_0(\emptyset)$, $Cp_0(\emptyset)$, $Cf_0(\emptyset)$.

Most of these rules are a formalisation of the centering theory adapted for generation. The main difference compared to the original centering theory is the explicit mention of communications in rules. For example, the continuation rule is simply expressed as $Cb_t(B) \wedge Cb_{t+1}(B) \wedge Cp_{t+1}(B)$ in the centering theory. However, it is unclear when we should consider that t becomes $t+1$. Most of the text analyses based on centering are made at the sentence level, i.e., t increases with each sentence, but some are made at the phrase level. By adding communications in the rules, we explicitly associate the increment in t with the output of a communication.

The no centre rule is new. This rule is necessary for text generation. In the case of text understanding, a sentence without backward-looking centre, such as the first sentence of a text, just sets things up but does not trigger any rules. This is why an anaphor in the first sentence of a text cannot be resolved by the centering theory since no rule can yet apply. However, for text generation, we need to trigger a rule to produce sentences without a backward-looking centre. This is the role of the no centre rule.

4.2.1 Ordering of the rules

Several rules may have their preconditions satisfied at the same time. We prefer then to apply the rule that maintains the focus if possible, or minimises its movement. Rules are therefore applied in the following preference order:

continuation > smooth-shift > retaining > rough-shift > no centre.

This is a variation on the original ordering where retaining is preferred to smooth-shift (Brennan et al. 1987). This change has been done in order to present new things as soon as possible. It means that we first try to find a communication to be output which would allow a “continuation” move. Then, if such a communication does not exist, we would try to find a “smooth-shift” communication and so on until a communication is found.

Preferring a smooth-shift to retaining introduces new things into the dialogue as soon as possible. This is equivalent to McKeown’s rules (McKeown 1985, p. 64).

An example of the use of this theory is given in section 5.

4.2.2 Translation rules

Because the focus rules we presented in section 4.2 are based on generic concepts such as backward or forward looking centres which are not necessarily used by the elicitation module, we need again to bridge the gap between these representations.

In the case of this theory, we need to define what the forward focus centres are and what the preferred focus centre is. Since this information is easily available from the messages exchanged by the elicitation module and the dialogue manager, the translation rules are particularly simple:

- $\forall t.(Cf_t(\mathcal{C}_u, X_1, \ldots, X_n) \ \wedge \ X \in X_1, \ldots, X_n \leftrightarrow subject(\mathcal{C}_u, X))$. Everything the communication mentions is a forward-looking centre.
- $\forall t.(Cp_t(\mathcal{C}_u, X) \leftrightarrow mainsubject(\mathcal{C}_u, X))$. The main subject of the communication is the preferred centre. The other forward-looking centres are not particularly ordered as they only present background information.

4.2.3 Surface phenomena

Example 3

S: Researcher set `rs1` is involved in research group `rg1`
Do you want *it* represented by a site?

The centering theory has been especially created in order to account for the use of pronouns. In our system, we make use of this capability to pronominalise some of the entities during the generation of communications and to resolve some anaphora during the parsing of sentences. In particular, we allow the backward-looking centre to be realised by a pronoun. This is the case in example 3 where the backward-looking centre of the second sentence is researcher set `rs1` since it was the first sentence’s preferred centre.

5 Example

In this section, we present how the dialogue manager and elicitation system interact to produce example 4. We describe how our system selects moves based on the focus theories and the translation rules presented in section 4, and we show how the specification and the focus spaces are modified. This is a simplified account of what really happens since our system deals with more entities and relations. Moreover, the mixed-initiative capability of the system is not shown. Nevertheless, it demonstrates the most important mechanisms.

Example 4

S1:	Research group rg1 is the research group.
S2:	Do you want a site presenting it?
U1:	Yes.
S3:	A site presents the research group.
S4:	Which site presents the research group?
U2:	A new site.
S5:	Site s1 presents the research group.
S6:	Do you want a home page describing site s1?
U3:	Site s1 is described by a new home page.
S7:	Home page hp1 describes site s1.
S8:	What is the title of home page hp1?
U4:	Research group.
S9:	Do you want a page linked from it?
U5:	Yes.
S10:	A page is linked from research group home page hp1.
S11:	Which page is linked from research group home page hp1?
U6:	A new page.

The two focus theories presented above are used to rank the communications. The global theory proposes an order which is refined by the local theory. The global theory has precedence over the local theory in the ranking process since it organises the dialogue at a higher level. The elicitation module starts working on the Domain Specific Specification Model described in figure 2. The initial Specification is empty. The Dialogue Manager uses the "translated" model presented in figure 5. A particular point of interest is the interaction between the dialogue manager and the elicitation system. The elicitation system influences the dialogue manager by adding communications to or removing communications from the communication pool (see section 3.2). The dialogue manager influences the elicitation system by deciding at which point in the dialogue the communications are output. This is different from systems where a Natural Language Interface is used to generate outputs and interpret inputs from users, but is still under the control of the reasoning system. In our system, the elicitation module constrains what can be output but, reciprocally, the choices made by the dialogue manager constrain what the elicitation module can work on.

We present the state of the system, i.e., the content of the specification and the content of the communication pools, after communications are output or interpreted. However, in order to simplify the presentation, we only present the changes made in the specification since most of its content stays the same. In particular, three kinds of changes can be made: (1) Assertions of things in the specification, e.g., changes from hypothetical to permanent status for things whose existence is confirmed by users, (2) Removal of hypothetical things from the specification when users deny their existence, (3) Addition of hypothetical things to the specification when the system believes they might exist.

The system is programmed in SICSTUS Prolog and runs on UNIX SUN workstations. It asks and answers questions in real time. The dialogue presented in example 4 is carried out between the system (S) and a human user (U).

Readers who are not interested in a detailed presentation of the system at work can safely skip this section. Section 6 will consider an evaluation of the dialogue manager.

5.1 Output S1 (Research group rg1 is the research group)

The system starts by creating all the mandatory entities and relations. This depends on the minimum cardinalities defined in the domain specific specification model used by the system (see table 2). In our case, this step involves the creation of a research group, a researcher set, a publication set and the `involve` relations between them. These creations are associated with the production of communications describing the changes made to the specification. Then, the system creates hypothetical entities and relations which represent possible changes to the specification. All the possible entities and relations which do not violate the constraints imposed by the domain specific specification model are created. These creations are associated with communications used to ask confirmation or denial of the proposed changes from users. The system state after these creations is presented in table 9. Given this initial state and the fact that no focus information is available to select the communication to output, the research group is presented by default. This is done by an ad-hoc rule, only used at the start of the dialogue. S1 is therefore produced. It is associated by default to a no change global focus move and a no center local focus move. The system state becomes that of table 10.

5.2 Output S2 (Do you want a site presenting it?)

In order to choose the next communication to output, the normal selection process is applied, now that focus information is available. The first theory to apply is the global focus theory.

A specialisation relation exists between `rg1` and both `i1` and `i2`. Direct relations exist between `rg1` and both `p1` and `p2`. This is shown in figure 5.

The only communications associated with a no change global focus move are communications `ask-creation(p1 | rg1, site)` and `ask-creation(p2 | rg1, page)`. These two communications are therefore passed to the next theory, the local focus theory. On the other hand, the communications which are associated with either an additive or a digressing move are delayed since a no change move is preferred over these moves. They will be output at a later stage, when the state of the dialogue is suitable. The two communications passed to the local focus theory are both associated with retaining local focus moves. The local focus theory cannot select one of them. The ranking process therefore stops with the two communications on-par. This is summarised in table 11. (Global and focus spaces are noted at the top of the table. Elements of no interest are replaced by underscores. The selected communication is set in bold

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
?entity(?s1, site)	rolefillervalue(i2, involved in, rg1)
?entity(?s2, site)	?rolefillervalue(p1, presenting, rg1)
?entity(?s3, site)	?rolefillervalue(p1, presented by, ?s1)
?entity(?pa1, page)	?rolefillervalue(p2, presenting, rg1)
?entity(?pa2, page)	?rolefillervalue(p2, presented by, ?pa1)
?entity(?pa3, page)	?rolefillervalue(p3, presenting, rs1)
relation(i1, involve)	?rolefillervalue(p3, presented by, ?s2)
relation(i2, involve)	?rolefillervalue(p4, presenting, rs1)
?relation(p1, present)	?rolefillervalue(p4, presented by, ?pa2)
?relation(p2, present)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	

Communication pool
present(rg1)
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
ask-creation(p1 | rg1, site)
ask-creation(p2 | rg1, page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_0 \{\}$

Centres
$Cf_0(\emptyset)$
$Cp_0(\emptyset)$
$Cb_0(\emptyset)$

Table 9: System initial state

Specification changes (since the initial state)
None

Communication pool
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
ask-creation(p1 | rg1, site)
ask-creation(p2 | rg1, page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_1 \{rg1\}$

Centres
$Cf_1(rg1)$
$Cp_1(rg1)$
$Cb_1(rg1)$

Table 10: System state (after S1)

face. The reasons for the focus moves of the selected communication are given

Communication	Global focus move $\mathcal{F}_1 =_1 \{rg1\}$	Local focus move $Cb_1(rg1), Cp_1(rg1), Cf_1(rg1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
ask-creation(p1 | rg1,site)	no change $dir_1(\_, p1, rg1)$	retaining $Cb_1(\_, rg1), Cp_1(\_, p1)$
ask-creation(p2 | rg1, page)	no change $dir_1(\_, p2, rg1)$	retaining $Cb_1(\_, rg1), Cp_1(\_, p2)$
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 11: Communication ranks (for S2)

Specification changes (since after S1)	Communication pool	Focus set
None	present(i1 | rs1, rg1)	$\mathcal{F}_1 =_2 \{rg1, p1\}$
	present(i2 | ps1, rg1)	Centres
	ask-creation(p1 | rg1, site)	$Cf_2(p1, rg1)$
	ask-creation(p2 | rg1, page)	$Cp_2(p1)$
	ask-creation(p3 | rs1, site)	$Cb_2(rg1)$
	ask-creation(p4 | rs1, page)	
	ask-creation(p5 | ps1, site)	
	ask-creation(p6 | ps1, page)	

Table 12: System state (after S2)

in each focus move cell.)

One communication is then selected at random by the selection process. If we assume that communication `ask-creation(p1 | rg1, site)` is selected, the system state becomes that of table 12. Because `rg1` is the backward-looking centre after S2 was selected and the local focus move is a retaining move, `rg1` can be realised by a pronoun.

5.3 Input U1 (Yes)

Since the user answers `yes` to the question, `answer(ask-creation(p1 | rg1, site), yes)` is added to the communication pool. Then, the eliciation module interprets this answer and the specification is modified to reflect it. In particular, relation `p1` and its associated roles are asserted in the specification; relation `p2` and its associated roles and communications are removed from the specification and the communication pool respectively. This shows why the dialogue manager cannot reliably plan the dialogue ahead since communications may be removed from the communication pool. A new presentation communication to inform the

Specification changes (since after S2)
Asserted:
relation(p1, present)
rolefiller(p1, presenting, rg1)
rolefiller(p1, presented by, ?s1)
Removed:
?entity(?pa1, page)
?relation(p2, present)
?rolefillervalue(p2, presenting, rg1)
?rolefillervalue(p2, presented by, ?pa1)

Communication pool	Focus set	Centres
present(i1 | rs1, rg1)	$\mathcal{F}_1 =_2 \{rg1, p1\}$	$Cf_2(p1, rg1)$
present(i2 | ps1, rg1)		$Cp_2(p1)$
present(p1 | rg1, site)		$Cb_2(rg1)$
ask-role-filler(p1 | presented by, site)		
ask-creation(p3 | rs1, site)		
ask-creation(p4 | rs1, page)		
ask-creation(p5 | ps1, site)		
ask-creation(p6 | ps1, page)		

Table 13: System state (after U1)

user of the changes made and a new question communication about `p1` role-filler are added to the communication pool. The resulting system state is presented in table 13. (We do not represent the users' answers in the communication pool.)

5.4 Output S3 (A site presents research group rg1)

The communication selection process is run once more. The global focus theory is first applied. Only communications `present(p1 | rg1, site)` and `ask-role-filler(p1 | presented by, site)` are associated with a no change global focus move. A direct relation exists for these two communications with `rg1` (since `p1` is in direct relation with itself and because of translation rule D3, Role = presented by, respectively). They are therefore selected and passed on to the local focus theory. The two communications are associated with a smooth-shift local focus move. This is presented in table 14. Since communication `present(p1 | rg1, site)` is a presentation communication whereas `ask-role-filler(p1 | presented by, site)` is a question communication, it is selected (see section 3.2). The new system state is presented in table 15.

Communication	Global focus move $\mathcal{F}_1 =_2 \{rg1, p1\}$	Local focus move $Cb_2(rg1), Cp_2(p1), Cf_2(p1, rg1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
present(p1 | rg1, site)	no change $dir_2(\_, p1, p1)$	smooth-shift $Cb_2(\_, p1), Cp_2(\_, p1)$
ask-role-filler(p1 | presented by, site)	no change $dir_2(\_, p1, p1)$	smooth-shift $Cb_2(\_, p1), Cp_2(\_, p1)$
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 14: Communication ranks (for S3)

Specification changes (since after U1)	Communication pool
None	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-role-filler(p1 | presented by, site) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_3 \{rg1, p1\}$

Centres
$Cf_3(p1, rg1)$ $Cp_3(p1)$ $Cb_3(p1)$

Table 15: System state (after S3)

5.5 Output S4 (Which site presents research group rg1?)

The question communication is then itself output with a no change, continuation move.

5.6 Input U2 (A new site)

Given the user's answer, a new site is created and added to the specification. A presentation communication presenting this change is added to the communication pool. The change also leads to the creation of hypothetical elements related to this new site and to the creation of new question communications. The new system state after the interpretation of input U2 is presented in table 16.

Specification changes (since after S4)
Asserted:
entity(s1, site)
rolefillervalue(p1, presented by, s1)
Added:
?entity(?hp1, home page)
?relation(d1, describe)
?rolefillervalue(d1, describing, s1)
?rolefillervalue(d1, described by, ?hp1)

Focus set
$\mathcal{F}_1 =_4 \{rg1, p1\}$

Centres
$Cf_4(p1)$
$Cp_4(p1)$
$Cb_4(p1)$

Communication pool
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
present(p1 | rg1, s1)
ask-creation(d1 | s1, home page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Table 16: System state (after U2)

5.7 Output S5 (Site s1 presents research group rg1)

Direct relations exist for the two new communications (since `p1` is in direct relation with itself, and `d1` is in direct relation with `s1` by rule D2, T_r = describe, $Role_1$ = describing, $Role_2$ = described by, rolecardmin(describe, describing, 1)). They can both be associated with no change global focus move. However, the presentation communication is associated with a continuation local focus move, since it is about `p1`, while the question communication is associated with a rough-shift. The presentation communication is therefore selected with a no change, continuation move. It is then output.

5.8 Output S6 (Do you want a home page describing site s1?)

The question communication is then selected with a no change, rough shift move and output.

5.9 Input U3 (Site s1 is described by a new home page)

Given the answer of the user, the hypothetical relation is asserted.

5.10 Output S7 (Home page hp1 describes site s1)

The selection process is run again for the home page which is an optional, dependent entity of the site. The state of the system after S7 is fully presented in table 17.

5.11 Output S8 (What is the title of home page hp1?)

Since home pages have titles, the elicitation system puts a question communication about the title in the communication pool. At the same time it creates a

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
entity(s1, site)	rolefillervalue(i2, involved in, rg1)
entity(hp1, home page)	rolefillervalue(p1, presenting, rg1)
?entity(?pa1, page)	rolefillervalue(p1, presented by, s1)
?entity(?hp2, home page)	rolefillervalue(d1, describing, s1)
?entity(?s2, site)	rolefillervalue(d1, described by, hp1)
?entity(?s3, site)	?rolefillervalue(l1, linked from, hp1)
?entity(?pa2, page)	?rolefillervalue(l1, linked to, ?pa1)
?entity(?pa3, page)	?rolefillervalue(l2, linked from, hp1)
relation(i1, involve)	?rolefillervalue(l2, linked to, ?hp2)
relation(i2, involve)	?rolefillervalue(p3, presenting, rs1)
relation(p1, present)	?rolefillervalue(p3, presented by, ?s2)
relation(d1, describe)	?rolefillervalue(p4, presenting, rs1)
?relation(l1, link)	?rolefillervalue(p4, presented by, ?pa2)
?relation(l2, link)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	

Communication pool
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
ask-value(hp1 | title)
ask-creation(l1 | hp1, page)
ask-creation(l2 | hp1, home page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_7 \{rg1, p1, s1, d1, hp1\}$

Centres
$Cf_7(d1, s1, hp1)$
$Cp_7(d1)$
$Cb_7(d1)$

Table 17: System state (after S7)

Communication	Global focus move $\mathcal{F}_1 =_7 \{rg1, p1, s1, d1, hp1\}$	Local focus move $Cb_7(d1), Cp_7(d1)$ $Cf_7(d1, s1, hp1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
ask-value(hp1 | title)	no change $dir_7(\_, hp1, hp1)$	smooth-shift $Cb_7(\_, hp1)$, $Cp_7(\_, hp1)$
ask-creation(l1 | hp1, page)	additive	
ask-creation(l2 | hp1, home page)	additive	
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 18: Communication ranks (for S8)

Specification changes (since after S7)	Communication pool
Asserted: attributevalue(hp1, title, "Research group")	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-creation(l1 | hp1, page) ask-creation(l2 | hp1, home page) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)

Focus set	Centres
$\mathcal{F}_1 =_8 \{rg1, p1, s1, d1, hp1\}$	$Cf_8(hp1)$ $Cp_8(hp1)$ $Cb_8(hp1)$

Table 19: System state (after U4)

hypothetical link with a page, since a home page can be linked to other pages. It also puts a question communication about the validity of this link.

The homepage is in a direct relation with itself (translation rule D1) whereas the link is in a specialisation relation (since there can be several linked pages, translation rule S1 applies, T_r = link, $Role_1$ = linked from, $Role_2$ = linked to, rolecardmax(link, linked to, ∞)). Because direct relations are preferred (no change moves are ranked higher than additive moves), the question about the title is selected and output. This is shown in table 18.

5.12 Input U4 (Research group)

The answer of the user is processed by the elicitation module once it has been asserted in the communication pool. The resulting system state is presented in table 19.

Specification changes (since after U4)
None

Communication pool
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
ask-creation(l1 | hp1, page)
ask-creation(l2 | hp1, home page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_9 \{rg1, p1, s1, d1, hp1\}$
$\mathcal{F}_2 =_9 \{l1\}$
controlling $_9(\mathcal{F}_1, \mathcal{F}_2)$

Centres
$Cf_9(l1, hp1)$
$Cp_9(l1)$
$Cb_9(hp1)$

Table 20: System state (after S9)

5.13 Output S9 (Do you want a page linked from it?)

Communications `present(i1 | rs1, rg1)`, `present(i2 | ps1, rg1)`, `ask-creation(l1 | hp1, page)` and `ask-creation( l2 | hp1, home page)` are all associated with an additive global focus move. There is no communication associated with a no change global focus move. These communications are therefore passed on to the local focus theory. Communications `ask-creation(l1 | hp1, page)` and `ask-creation(l2 | hp1, home page)` are associated with a retaining local focus move. They are therefore preferred over communications `present(i1 | rs1, rg1)` and `present(i2 | ps1, rg1)` which are associated with a no centre local focus move. Because the two communications are question communications, we do not make any difference between them. One of them is therefore selected at random. If we assume that communication `ask-creation(l1 | hp1, page)` is selected, S9 is output. A new focus space is created by the action of the additive global focus move. It will contain everything related to the new page. The new system state is presented in table 20. Because the home page is the backward-looking centre of the current sentence and was the backward-looking centre of the previous sentence too, it is realised by a pronoun.

5.14 Communications U5 (Yes) to U6 (A new page)

What happens in the rest of the example until input U6 is equivalent to what happened between input U1 and input U2. The main difference is that all the new entities spoken about are put in the new global focus space $\mathcal{F}_2$. Note that after U5 the title of the home page is used in the generation process. This makes it easier to recognise the home page spoken about since its name is more explicit than a simple identifier.

5.15 Summary

As can be seen in this example, global and local focus rules constrain the evolution of the dialogue. For example, global focus rules delay the presentation of the researcher set and the presentation of the pulication set while the research group is in focus. Then, when several additive moves are possible, local focus rules constrain the dialogue manager to output the communication which is more related to the current topic.

The dialogue depends on the communication selection made by the dialogue manager. However, this selection is also constrained by the availability of communications in the communication pool and therefore by the elicitation module. If the elicitation module follows a strict and narrow way of working, always producing a unique communication, then the dialogue manager becomes irrelevant. On the other hand, putting several communications in the communication pool entails that their treatment can be deferred for an unknown amount of time, until the dialogue manager decides to output them. As a result, the elicitation system should balance the advantages and drawbacks of putting too few or too many communications in the communication pool, and the dialogue manager should balance the advantages and drawbacks of more or less delay in the output of communications. This kind of interaction between a reasoning module (here the elicitation module) and the dialogue manager is novel. Usually, one of the modules is charge of the process. For example, in information retrieval dialogues, the dialogue manager organises the interaction with the users. In task oriented settings, the dialogue is usually directed by the reasoning module. In our approach, both modules cooperate to produce a dialogue which is suitable for the reasoning module and the system users.

6 Evaluation

Numerous pieces of work make use of focus theories. The theories are sometimes modified to achieve specific aims (e.g, Mittal et al. (1998); Huang (1994)). However, most of these approaches do not precisely evaluate what the contributions of the theories are. The fact that they provide improvements in dialogue quality is assumed. In this section, we present an evaluation of the impact of using focus theories. This evaluation was made during the development of the system, helping decide to which aspect of the system efforts should be devoted. We describe what the contributions of the different theories are, with an emphasis on the theories presented in section 4. The evaluation strategy we followed can be formalised. We present its general form in section 7.

We first describe the setting in which the evaluation has been done and the resulting constraints that applied to the evaluation (Sparck Jones and Galliers 1996).

In our case, we wanted to see how the focus theories influence the capability of the system to produce coherent dialogue. As shown in figure 1, the focus theories intervene, through the dialogue manager, by selecting a communication from the communication pool. This communication is then directed to the Natural Language Interface (NLI). Since we wanted to test the theories in the same conditions, we chose to use the same elicitation module and the same NLI for all tests. The elicitation system is a loose one, i.e., putting as many communications in the communication pool as possible (see section 3). It offers many choices for selection. The NLI is based on templates. These can be modified to allow changes such as the use of pronouns depending on the focus information available. The resulting evaluation can be seen as a black-box evaluation, where the influence of the theories are evaluated through the behaviour of the whole

	No management[2]	One focus theory	Two focus theories
Evaluations	2, 3, 5	15, 10, 6	10, 12, 12, 10, 8

Table 21: First evaluation
Evaluations (vary from 0 (very bad) to 16 (excellent))

system.

We considered that users of the system are accustomed to using computers but not our system. Users who had no linguistic knowledge were given short explanations on the evaluation they had to perform in order to ensure they understood exactly what was asked in the evaluations.

Using a focus theory is compared to having no management, using other theories and using other management methods (not focus based). It is indeed not useful to implement complex focus theories if they do not provide improvements over simpler methods. The use of the theories should make a real change, so that users can feel the difference from other dialogue management strategies.

6.1 First evaluation

The aim of the first evaluation was to determine if focus theories have an influence on dialogue quality. In order to evaluate this, we defined a criterion of dialogue quality based on dialogue properties such as ease of understanding and well-structuredness. The evaluation consisted in having users use our system with and without dialogue focus rules, and then filling in a questionnaire about the dialogue they had had. The criterion was then calculated by summing up the answers in the questionnaire, each question counting from 0 up to 4 depending on the answer (the higher the user's satisfaction, the higher the mark), for a maximum total of 16.

Eight users were asked to enter a given specification. Users had mostly to answer questions as in the example presented in section 5. Users could also "pass" them, i.e., decline to answer them at that point. They could also ask to see the current state of the specification. Other mixed-initiative possibilities were limited.

Seven of the users had never used the system before. We obtained eleven dialogues (some users producing more than one dialogue) and their evaluation. The results are presented in table 21. The experiment lasted less than two hours, including a presentation of the system to the new users. We found that there was a statistically significant difference between the evaluation of a random dialogue, i.e., a dialogue where communications are picked at random in the communication pool, and the evaluation of a dialogue using at least one focus theory (Student test with null-hypothesis 'no difference in evaluation means', $t = 4.15$; degree of freedom, $df = 9$; probability of error in rejecting null-hypothesis, $p = 0.002$).

We also tested whether the improvements in perceived dialogue quality were related to better specifications. Therefore we searched whether the dialogue

[2] No focus theories were used. Presentation communications were preferred over question communications but otherwise communications were selected at random from the communication pool (see section 3.2).

evaluation was related to the number of errors made while entering the given specification. Results point out that there may be a reduction of errors with better dialogues. However, there were not enough data to get satistically significant results (correlation coefficient, r = -0.1926; error on r, $\sigma_r = 0.2$; Student test with null-hypothesis 'r = 0', t = -0.95; degree of freedom, df = 9; probability of error in rejecting null-hypothesis, p = 0.3657).

6.2 Second evaluation

Although the first evaluation let us know that focus rules may be useful, it does not show that they perform better than other dialogue management strategies such as following the elicitation module order, i.e, always selecting the latest communication added to the communication pool. Testing these possibilities would require a heavy investment in users' evaluation of the system. In order to reduce this investment while being able to test other dialogue management strategies, we designed an automated evaluation procedure.

6.2.1 Automation

The aim of the automation is to replace the real users with automated users, while obtaining an evaluation that remains faithful to the real users' perception of the dialogues. The automation of the evaluation is composed of two steps as shown in figure 6. Each step consists in automating a part of the evaluation process. Each step provides gains but involves possible losses. Counter-measures are taken to limit these losses.

The first step consists in finding a metric, based on easily measurable dialogue properties, which is correlated with the evaluation given by users. The idea is that the metric should allow an easy and clear separation between the strategies which provide improvements and those that do not. Using this metric we are able to evaluate automatically the quality of a given dialogue. Possible problems are a loss in accuracy and in faithfulness to the initial evaluation. To counterbalance this loss, we searched for a metric well-correlated with the evaluation and psychologically motivated. By ensuring that the metric is psychologically motivated, we try to screen out metrics that could be highly correlated with the evaluation just by chance, because of the small number of real evaluations. Several metrics, often proposed in the literature as being related to dialogue quality, were recorded during the users' evaluation. We found that some obvious choices were not satisfactory in our case. For example, the length of the dialogue was not well correlated to the evaluation (r = 0.4019). The number of passed communications (r = -0.1207) and the number of times users asked to see the specification (r = -0.3595) were not good metrics either.

We found that one of the metrics accounted for 65% of the evaluation mark (correlation coefficient, r = 0.8419; error on r, $\sigma_r = 0.2$; Student test with null-hypothesis 'r = 0', t = 4.08; degree of freedom, df = 9; probability of error in rejecting null-hypothesis, p = 0.003). This metric is based on dialogue coherence: communications related to the previous ones increment the metric[3].

[3]More precisely, a communication was considered related to the previous ones if (1) it was a presentation communication or (2) its main subject was among the last seven subjects spoken about. A bonus was accorded if the communication main subject was among the last two subjects spoken about (in most cases, this means that the main subject was related to

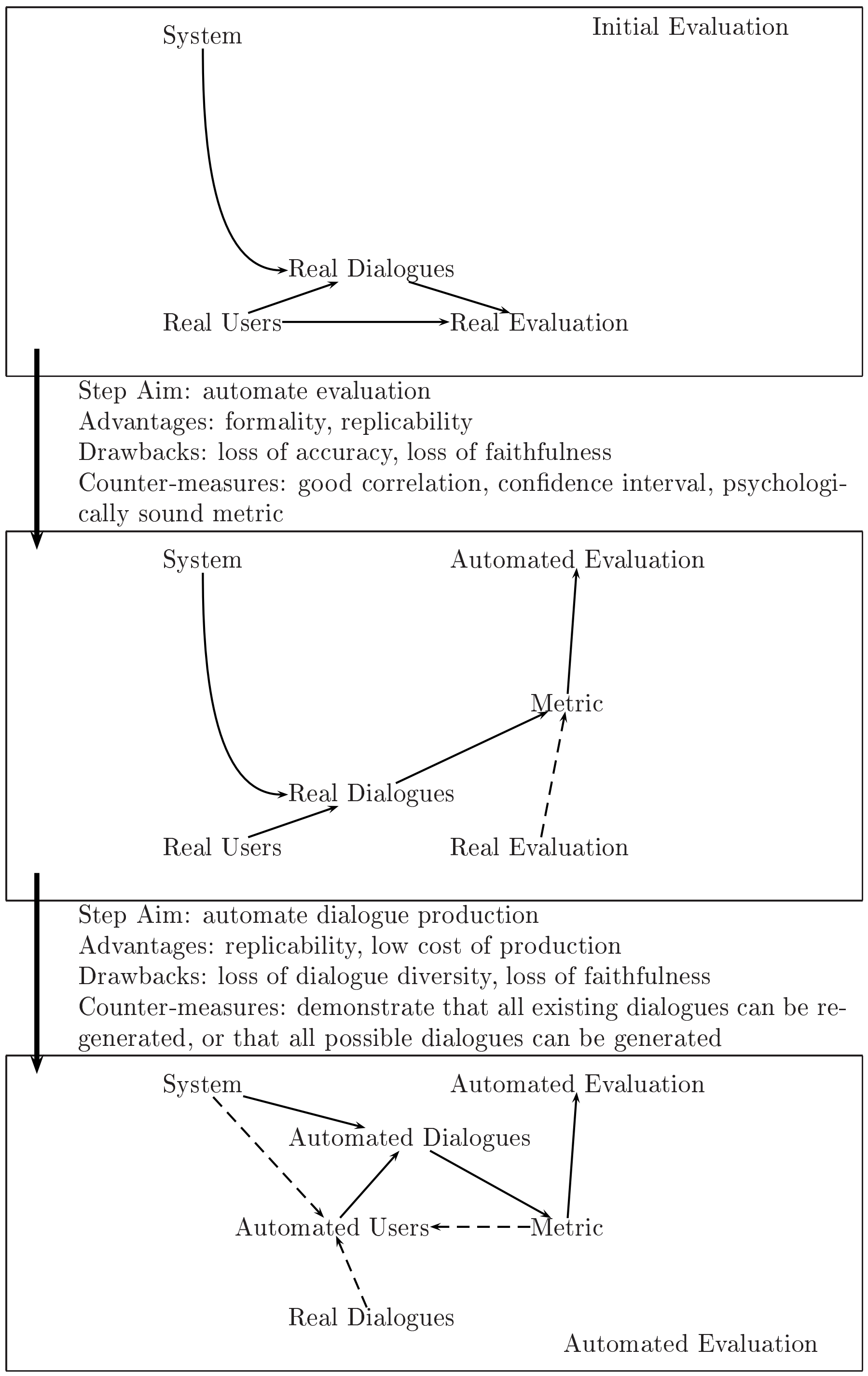

Figure 6: Automated evaluation

Solid lines show how the evaluation is carried out. Dashed lines show how automated components are obtained.

This metric seems psychologically possible given human beings' memory limitation (Miller 1956). If the main subject of a communication is not fresh in the users' memory, users may have difficulties in interpreting the communication. Moreover, the fact that this metric is based on dialogue coherence encourages us to investigate the role of focus theories more precisely. Other metrics, such as metrics combining several aspects like the number of passed communication and the number of times users asked to see the specification, were tested. Some had higher correlation coefficient. However, they were either not statistically significant or were not psychologically sound, i.e., their results could not be easily explained. Therefore, we chose the metric based on dialogue coherence to approximate the users' evaluation.

The second step consists in producing dialogues automatically with different versions of the system and comparing them using the metric. This is done by writing automated users. The aim of these programs is to approximate the dialogues of real users well enough that the metric will produce useful evaluations. The metric divides dialogues into equivalence classes (dialogues having the same metric value at a given confidence level). Therefore, these users do not need to imitate all aspects of the real users but only produce dialogues of the same equivalence class as real users. More accurate automated users would not always provide much more accurate evaluations since the metric would often level the results off. Therefore we wrote simple and cost-effective automated users. These users know about a target specification and answer the questions asked by the system about it. In order to do this, they map the entities mentioned in the dialogue with their counterparts in the specification. The mapping is done when presentation communications are processed. For example, if a research group is mentioned in the dialogue and a unique research group exists in the target specification, the two entities will be mapped. This means that the identifier of the research group in the dialogue, e.g., `rg1`, is associated with the identifier of the research group in the automated user's internal specification. This association is stored in a "Mappings" list. Questions can then be answered by mapping the elements of the questions to their internal counterparts and then searching for an answer in the target specification, simulating what might have been the answers if human users had been responding. If presentation communications cannot be processed, e.g., because the system has not already presented things needed to interpret them, they are stored for later processing. If questions cannot be answered, they are passed.

A possible problem with the automation is the loss of dialogue diversity due to simple automated users. We ensured this was not the case by verifying our automated users could generate dialogues equivalent in length and structure to the ones produced by real users. In order to achieve this, we had to limit the memory span of our automated users so that they would forget things as our human subjects were observed to do. We approximate this by making the simulated users keep a list of things in focus. The list, called the "Focus" list, has a limited storage capacity. New things put in the list may therefore displace those already in the list, on a First-In-First-Out (FIFO) basis. If a question is about things that are not in the list, the automated users will answer or pass the question with some probability. The algorithm driving the automated users is described in table 22.

---

the latest communication).

```
Unprocessed Mappings ← ∅ {contains the presentation communications that could not be processed immediately}
Mappings ← ∅ {contains the mapping between dialogue entities and their internal counterparts}
Focus list ← ∅ {contains the things spoken about recently and acts as a FIFO}
while there is a communication to process coming from the dialogue manager do
  if this is a presentation communication then
    if it can be interpreted using Mappings then
      add any new mappings to Mappings
      add its subjects to the focus list
      repeat
        select a communication from Unprocessed Mappings
        if it can be interpreted using Mappings then
          add any new mappings to Mappings
          remove the communication from Unprocessed Mappings
        end if
      until no communication can be removed
    else
      add the communication to Unprocessed Mappings
    end if
  else
    if it can be interpreted using Mappings then
      if its subjects are in the focus list then
        answer the question
        add any new mappings to Mappings
        add its subjects to the focus list
      else
        if a random function signals the choice to answer then
          answer the question
          add any new mappings to Mappings
          add the subjects to the focus list
        else
          pass the communication
        end if
      end if
    else
      pass the communication
    end if
  end if
end while
```

Table 22: Automated users' algorithm

Finally, we can run the system with the automated users and obtain an evaluation of new dialogue management strategies. The results given by this approach are not very accurate since the use of a metric and automated users introduces approximations. However, these losses in accuracy can be partially evaluated and counterbalanced. For example, by considering confidence intervals rather than point evaluations, we can get a better opinion on the real differences between strategies. Confidence intervals can be computed by running the automated users several times with the same dialogue management strategy. This enables us to determine the uncertainty on the automated evaluation due to the random element in the automated users' dialogues. We can then combine this result with the uncertainty on the metric (due to the imperfect correlation with a real evaluation) to obtain an overall uncertainty measure for the automated evaluation.

Two main assumptions are made during the automation process:

1. A well-correlated and psychologically sound metric can be found. Numerous metrics have been studied and evaluated in the literature. However, this may still be difficult for dialogue aspects on which little research has been carried out.

2. Automated users imitating real users can be written. This is usually possible since the system under study can only accept a limited number of answers. However, this may still be difficult to achieve if, for example, the users need a lot of common-sense knowledge.

Not all new dialogue management strategies can be tested in this way. The use of a metric puts constraints on the type of dialogues that can be evaluated. For example, the length of the dialogue was shown not to be a good metric in our case. However, this metric could be good for other dialogues, e.g., if the dialogue manager's main role was to remove communications rather than ordering them. If the type of dialogues evaluated changes, a new metric should be used. The same pertains to automated users. These users have been written to have a dialogue with a particular type of dialogue manager. If the latter changes too much, automated users may not be able to deal adequately with it. It is therefore important to compare dialogue managers that are not "radically" different.

Automated users implementing this algorithm have been written in SICSTUS Prolog and run on UNIX SUN workstations. They can communicate with the dialogue manager thus producing dialogues automatically.

6.2.2 Results

Using the metric and the automated users, we were able to test if other dialogue management strategies that were not tested during the first evaluation could also provide improvements in dialogue quality. One such strategy was to simply follow the elicitation module order (in effect by-passing the dialogue manager and considering the communication pool as a FIFO store). This strategy turns out to perform slightly better than focus theories. This surprising result comes from the way our elicitation module works. By always outputting all the questions it can, it effectively implements a strategy that has some resemblance with the global and local theories mixed together (although the jump between global focus space is more random and local coherence is not strictly enforced). This would not be true of elicitation system in general, which by default do not manage dialogues. We also tested the theories presented in section 4 separately. The evaluations are shown in figure 7. (The theory presented in section 4.1 is

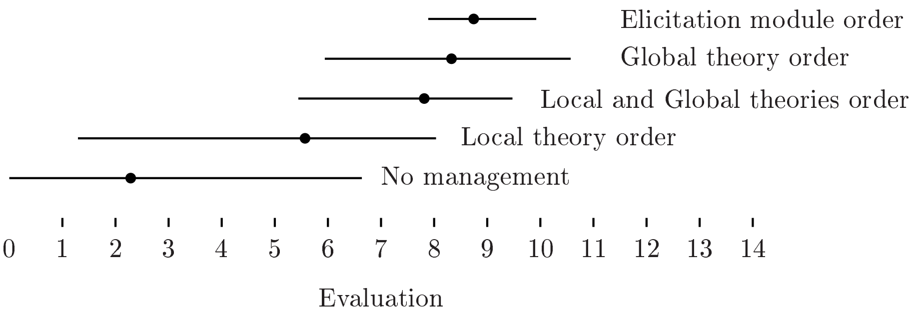

Figure 7: Evaluations (vary from 0 (very bad) to 16 (excellent)) and 90% confidence intervals

called "global" in the figure and the theory presented in section 4.2 is called "local".) The local focus theory, although better than the no management policy, performs less well than the global or elicitation order strategies. This could be due to the fact that a local dialogue management leads to "spaghetti" dialogues (Sibun 1992) which are not optimal in our case.

Based on these results, three strategies were selected for further evaluation: elicitation module order, global focus theory and the mix of global and local focus theories. The strategies were selected because they form the best cluster in the evaluation and we have found reasons to reject the other strategies.

6.3 Third evaluation

In the following evaluation, we evaluated the three selected theories without developing new ones. (The local theory has been dismissed by the previous evaluation but is added as a base case.) We did not search for a correlated metric (although we recorded metric values in case of further development) and we did not develop automated users.

We tested if the strategies were ensuring good global coherence. We tested this hypothesis by asking people to read transcripts of dialogues generated with and without dialogue strategies and asking them to compare the dialogues pair-wise based on their coherence.

The evaluation involved six persons and lasted 90 minutes in total. Six dialogues were evaluated, resulting in 36 pair-wise comparisons. (Each participant compared four dialogues.)

Pair-wise comparisons can be transformed in an overall ranking of the alternative dialogues (Saaty 1990). This is done by computing the principal eigen vector of the comparison grid. Its coordinates give the relative importance of the different strategies. For example, a theory with a value of 60 is considered three times more globally coherent than a theory with a value of 20.

The results are presented in table 23 (the local theory, typeset in italics, was

Local and global theories order	33
Global theory order	47
Elicitation system order	12
Local theory order	*8*
	100

Table 23: Theory ranking by global coherence

Local and global theories order	50
Global theory order	37
Elicitation system order	*13*
	100

Table 24: Theory ranking by local coherence

added as a base case, consistency ratio = 0.0818[4]). They show that following the elicitation system orders performs poorly compared to using a focus theory (3 to 4 times worse than a focus theory). This strategy is therefore abandoned. The results also confirms that local theory is performing badly as was predicted in the previous stage. The local and global theories are performing less well than the global theory alone because the local theory smooths the transitions between focus spaces. Therefore the dialogue seems less well divided than with the global theory only.

6.4 Fourth evaluation

Finally we tested the two remaining theories' capability to improve local coherence. We tested this hypothesis with the same kind of experiment as in the previous evaluation.

The results are presented in table 24 (the elicitation system order, typeset in italics, was added as a base case, consistency ratio = 0.1664). The theory mix consisting of a local and a global theories is performing better than the global theory alone. However, the difference is not big. This can be explained by the small size of the focus spaces. Since just a few communications are output in each space (around 4 on average), a suboptimal coherence strategy does not decrease the perceived local coherence too much.

6.5 Summary

From this analysis we can conclude that, in our domain, the local and global theory mix and the global theory on its own provide improvements in dialogue quality and especially in dialogue coherence. The global theory performs better on global coherence as can be expected and the theory mix performs well on local aspects. It should be noted that the local theory which seemed a potential strategy was eliminated quickly in the development process. The elicitation order, which was not at first considered, was introduced at a very low cost in the evaluation process but was later dismissed for its poor performance in

[4]A consistency ratio less than 0.1 indicates reliable results.

global and local coherence. All the evaluations needed to develop and evaluate the system took only 25 person-hours on aggregate.

This analysis was performed on a limited domain. However, the evaluation was not dependent on the domain. We therefore believe that the results have a wider application than the domain used here.

7 Development

In this section we show how we integrated the automation of the evaluation we used to get the results presented in the previous section into a general development method. We used this development method to produce the theories and results presented above. This method could be the basis for a new and generally useful development approach for interactive research systems.

Because there is not yet a clear method for the development of dialogue managers, an iterative approach using evolutionary prototypes seems adequate (Partridge 1992; Sommerville 1995). Each iteration is a step towards the final program. However, not going astray is a major difficulty. Evaluating whether a dialogue management strategy is beneficial or not, in terms of performance but also in terms of cost-effectiveness, is therefore important.

Evaluation of dialogue management strategies is a difficult task. Several sources can influence the dialogue beside the management techniques, and dialogue can be evaluated along several dimensions. Many current approaches to evaluation require heavy involvement from users (Sikorski and Allen 1996; Walker et al. 1997a). This makes them difficult to use during development. Some are based on automated users:

- Walker (1996) uses automated users whose dialogue strategies and memory and reasoning capabilities can be adjusted. Communication and reasoning have an associated cost. Dialogue performance is then calculated by considering the overall cost of the dialogue and how successful it was. This framework is useful for studying the theoretical implications of dialogue strategies depending on cognitive constraints (Hanks et al. 1993). However, it is unclear what the relation is between the dialogue performance computed and a real dialogue evaluation. (For example, counting the numbers of utterances may be less representative of the dialogue quality than of the elicitation system's capability to guess some of the specification.) Therefore, the application of this framework to dialogue quality evaluation is limited.

- Eckert et al. (1997, 1998) also uses automated users whose dialogue strategies can be adjusted. The adjustments are made by studying a corpus of existing dialogues. Dialogue quality is then calculating by using a metric which is chosen to be well correlated with real users' evaluation (Walker et al. 1997b). This framework is useful to test different dialogue strategies on a particular aspect of the dialogue and for specific users. However, since only one metric is used, it is not clear how to test the dialogues on different aspects, such as global and local coherence. It is also unclear if investing a lot of effort in accurate automated users is worthwhile when the dialogue manager is undergoing modification, which is the case during development. New automated users may be needed if the

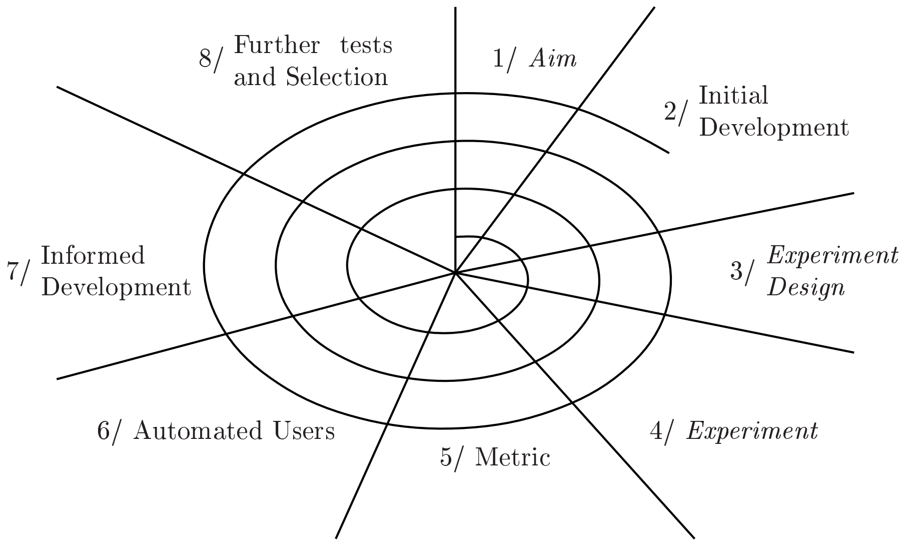

Figure 8: Development and Evaluation Cycle

dialogue manager evolves too much. It may therefore be better to write simpler automated users adapted to the metric used to test a particular aspect of the dialogues. The application of this framework during the development phase where the dialogue manager evolves and several aspects of the dialogues needs to be evaluated seems therefore limited.

- Ishizaki (1997) also makes use of automated users to evaluate the conditions under which mixed-initiative dialogues are more effective than non mixed-initiative dialogues. In this case, a simple metric, consisting of the number of characters of the dialogue, is used. The automated users are also simple and only roughly based on existing dialogues. However, a mathematical model gives a sound theoretical basis to the study and the simulation is used to confirm the theoretical findings. Unfortunately, such an approach is difficult to apply in our case since the notion of coherence is difficult to formalise and to study mathematically.

The main objective in our approach is to provide quick but faithful evaluations on particular aspects of the system. This is done by a light-weight evaluation process. Compared to the last two approaches presented above, we make fewer assumptions about the possibility of studying the dialogues statistically or analytically. This enables us to deal with more types of dialogues. On the other hand, we are not able to benefit from proven mathematical results, which means we have to ensure the correctness of our evaluation by repeating experiments. The process can be used while developing the system and can provide guidance on which aspect of the system should be worked on. The process is based on a spiral model of software development (Boehm 1987) which is well suited to our case. It is composed of a cycle shown in figure 8 and described below. Each turn of the cycle represents the development and evaluation of one aspect of the system. The radial movement therefore represents the amount of testing carried out on the system.

1. The first step consists in defining the aim of the current cycle turn. This step has one of two results:

    - The cycle is stopped. The system has reached its objective or there is no cost-effective way of making progress.
    - A possibility of improvement is identified. Knowing which improvements should be attempted and tested first depends on the final requirements for the system and on the perceived risk in achieving some progress. More important and more risky aspects of the system should be considered first. The constraints under which the improvement should be obtained and a testable criterion are defined. The improvement to be achieved is expressed in terms of that criterion. The aim of this criterion is to allow the partition of the possible additions to the system into those that improve the performance of the system in the way defined in the preceding step and those that do not. (We did this by defining a criterion for dialogue quality.) This ensures that the improvement can be measured and verified (Sommerville and Sawyer 1997, good practice 8.7).

      Application: In our case, in the first turn of the cycle we considered whether dialogue management was providing improvements in dialogue quality. In the second turn, we studied the effect of management on global coherence and, in the third, we studied the the effect of management on local coherence. Finally, we stopped in the fourth turn.

2. Next, development is carried out to provide the improvement wanted. Several competing strategies can be used to achieve this goal.

    Application: In our case, we developed several strategies, such as those based on local and global focus theories.

3. During this step, we design the evaluation that will be carried out in this turn of the cycle. Users' evaluation is necessary since there is usually no other way of knowing if a system is performing correctly: " in general, the straightforward notion of decidably correct or incorrect system behaviour even for specific inputs is difficult to pin down for many AI problems" (Partridge 1992, p30–31). We found that two kinds of experiments are especially useful for evaluation:

    - Interactive experiments where users play with the system and then answer a questionnaire. In general, this is a cost effective way of obtaining subjective data. This is all the more the case here, since we test only a small aspect of the system defined in step 1.
    - Static experiments where users do not directly interact with the system but judge some of its outputs. Comparison techniques (e.g., pair-wise comparison (Karlsson 1996)) can be used, usually reducing the amount of evaluation to be done.

Application: In our case, we chose an interactive experiment in the first turn of the cycle. We then chose static ones for the second and third turns.

4. The next step consists in the actual evaluation of the system by users. This is done by asking users to complete the experiment designed in step 3. At the same time, dialogues are recorded and metrics are measured. This step provides two results:

   - It enables us to know if there is a need for further evaluation. If no improvements can be gained, then there is no need to go further.
   - Assuming that there are improvements, we can automate the evaluation. This is the role of steps 5 and 6.

   Application: In our case, we found that some management strategies had an effect on the aspect we were studying.

5. A major problem in obtaining a cost-effective development method is the need to automate the evaluation step because evaluation by users is time-consuming. As explained in section 6.2, if we can find a metric correlated with the results found in step 4, then we can use this metric in place of the questionnaire given to the users. The idea is that the metric should allow an easy and clear separation between the strategies which provide improvements in dialogue quality and those that do not. Finding this metric is a difficult part of the process. Metrics should be kept as simple as possible, especially during the first cycles of the approach. Metrics should be based on dialogue properties and not on properties of the particular strategies used to generate the dialogues, since the comparison of the internal working of different theories could be difficult or even impossible. Note that the metrics need not explain the results but just be correlated with them. (In our case, the search for a metric was based on proposals made in the literature.) The difference between this metric and the criterion defined in step 1 is that the metric is measurable, i.e., is objective, whereas the criterion is testable, i.e., depends on users' evaluation.

   Application: In our case, we found a metric during the first turn of the cycle that accounted for 65% of the users' evaluation. We did not search for a metric during the next turns as we were not developing new strategies.

6. The optional step consists in writing automated users. It should be performed when users are directly experimenting with the system. Otherwise, it can be ignored. The aim of these programs is to approximate the dialogues of real users for the metric of step 5. Knowing which characteristics of the dialogue should be replicated by the automated users makes it easier to program them without having big corpora for evaluation. Once such automated users have been written, we can use them in place of real users. This is a big advantage as real users do often not have time to test a system under development. In particular, this enables us to test strategies that were not evaluated by real users (for example because they were developed later). This is the aim of the next step.

   Application: In our case, we developed, during the first turn of the cycle, simple automated users answering or passing the system's communication.

		Cycles			
		1	2	3	4
Steps	1	Issue: quality improvements	Issue: global coherence	Issue: local coherence	Stop
	2	Dialogue managers development (e.g., with local and global focus rules)			
	3	Interactive experiment design	Static experiment design	Static experiment design	
	4	Experiment	Experiment	Experiment	
	5	Metric			
	6	Automated users development			
	7	Further development and debugging			
	8	Further evaluations and selection	Selection	Selection	

Table 25: Applying development and test cycle in our approach

7. In the next step, new strategies are developed by taking into account the insights provided by the evaluation. This informed phase of development is better focused on the problem and should result in better strategies.

   Application: In our case, this step was performed during the first turn of the cycle. It mainly consisted in debugging the dialogue management strategies.

8. In this last step, we use the automated users of step 6 to produce dialogues and we evaluate them using the metric of step 5. We can therefore test different dialogue management strategies developed in the previous step at a minimum cost. If we want to test a new strategy, we can do it without requiring real users evaluation. Care should be taken, however, not to tailor the strategies to the particular tests carried out. Separating training and evaluation data is a good way of ensuring that the new developments are well-tuned but not over-specified for the tests used.

   Once strategies have been selected for further development, they are fed back to step 1 and the cycle recurs.

   Application: In our case, we tested several new strategies (or strategy mixes) during the first turn of the cycle. During all turns we also dismissed poorly performing dialogue strategies.

Not all steps in the cycle are performed at each turn. A full turn is carried out during development. However, during evaluation only, i.e., when no new strategies are created, the cycle is reduced to steps 1, 3 and 4 (typeset in italic in figure 8). This is particularly the case towards the end of development when the emphasis changes from development to testing. It is a good idea to terminate the process by some pure evaluation turns (as was done in our work as shown in table 25).

It should be noted that each new cycle is not necessarily concerned with a subset of the previous turns' concerns. An orthogonal issue can be dealt with. However, because dialogue management strategies are dismissed as soon as possible, more important issues should be dealt with first. Then, only the most important strategies will be ranked during the next turns. If a minor issue is dealt with first, an important strategy could wrongly be eliminated. This is why we tested our strategies on issues of decreasing importance from first to fourth evaluation.

The evaluation approach presented here can be used during development. The low cost of this method enables us to screen many different dialogue management strategies to pick out the best. These strategies can then be subjected to more in-depth evaluation. However, it can also be used to test fully-blown systems. In this case, each cycle test a particularly important point of the systems so that their strength and weaknesses are evaluated. The evaluation stops as soon as one of the systems can be selected.

8 Conclusion

The main aim of this paper was to present a dialogue manager organising dialogues between a reasoning tool and its users. We chose to write a dialogue manager based on focus rules and to embed it in a system eliciting requirements. We started the paper by explaining why we believe focus rules are well-adapted for organising the kind of dialogues considered here. Then, we presented the architecture of our system. We described the dialogue manager, and we showed how it communicates with the requirements elicitation module. We then presented the formalisation of two focus theories. Because the theories are formalised, we have been able to study some of their properties. We have also been able to code them. We stressed the fact that these theories need domain knowledge to operate. In order to reduce the amount of engineering work required to provide this knowledge, we showed a way of re-using the knowledge used by the reasoning module. We did this by writing "translation rules". These rules bridge the gap between the representation frameworks used by the dialogue manager and the elicitation module. We then analysed a sample dialogue produced by our system. We stressed the fact that the elicitation module and the dialogue manager cooperate to produce a dialogue, which is suitable for reasoning and easily understood by the system users. We have then evaluated the focus theories used in our system. We found that they provide improvements in dialogue coherence, except for the local theory on its own. It remains to be seen, however, whether these improvements translate into better specifications. In order to perform the evaluation, we have developed an innovative way of automating its process. Finally, we presented a generalisation of our evaluation strategy suited for incremental development.

Although we developed our system for a specific application, we believe that it is representative of a class of applications where reasoning and communication play important roles. In particular, it seems that similar methods of dialogue control could be used in domains where the dialogues are not completely determined by the reasoning tools, but are constrained enough to make automated focus control tractable. By taking into account rules of dialogue coherence, our approach could provide improvements in the way users interact with those tools. The evaluation method we propose seems also applicable in a wider context. By enabling developers to evaluate the dialogue manager at a low cost, it should allow them to fine tune the manager more easily.

References

William J. Black. Acquisition of conceptual data models from natural language descriptions. In *Proceedings of the 3^{rd} European conference of the ACL, Copenhagen, Denmark*, pages 241–248, April 1987.

Barry W. Boehm. A spiral model of software development and enhancement. In Richard H. Thayer, editor, *Software Engineering Project Management*, pages 128–142. IEEE Computer Society Press, 1987. Reprinted in Thayer and Dorfman (1997, p. 416–430).

Susan E. Brennan, Marilyn Walker Friedman, and Carl J. Pollard. A centering approach to pronouns. In *Proceedings of the 25^{th} Annual Meeting of the Association for Computational Linguistics, Stanford University, Stanford, California, USA*, pages 155–162, 1987.

Peter Pin-Shan Chen. The entity-relationship model – Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, March 1976.

Peter Pin-Shan Chen. English sentence structure and entity-relationship diagrams. *Information Sciences*, 29:127–149, 1983.

Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors. *Intentions in Communication*. The MIT Press, 1990.

Robin Cohen. A computational theory of the function of clue words in argument understanding. In *Proceedings of the 22^{nd} annual meeting of the Association for Computational Linguistics, Stanford University, California, USA*, pages 251–258, July 1984.

Robin Cohen. Analyzing the structure of argumentative discourse. *Computational Linguistics*, 13(1–2):11–24, 1987.

Nancy J. Cooke. Varieties of knowledge elicitation techniques. *International Journal of Human-Computer Studies*, 41:801–849, 1994.

Alan Davis. A comparison of techniques for the specification of external system behavior. *Communications of the ACM*, 31(9):1098–1115, September 1988.

Alan Davis, Scott Overmyer, Kathleen Jordan, Joseph Caruso, Fatma Dandashi, Anhtuan Dinh, Gary Kincaid, Glen Ledeboer, Patricia Reynolds, Pradid Sitaram, Anh Ta, and Mary Theofanos. Identifying and measuring quality in a software requirements specification. In *Proceedings of the 1^{st} International Software Metrics Symposium*, pages 141–152, 1993. Reprinted in Thayer and Dorfman (1997, p. 164–175).

Alan M. Davis. *Software Requirements: Objects, Functions, and States.* Prentice Hall, revised edition, 1993.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. User modeling for spoken dialogue system evaluation. Technical Report 97.33.1, AT&T Labs Research, 1997.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. Automatic evaluation of spoken dialogue systems. Technical Report 98.9.1, AT&T Labs Research, 1998.

Joseph A. Goguen and Charlotte Linde. Techniques for requirements elicitation. In *Proceedings of the* 1^{st} *IEEE international symposium on Requirements Engineering, Los Alamitos, USA* IEE (1993), pages 152–164.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226, 1995.

Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

Steve Hanks, Martha E. Pollack, and Paul R. Cohen. Benchmarks, testbeds, controlled experimentation, and the design of agent architectures. *AI Magazine*, 14(4):17–42, 1993.

J.-P. Haton, N. Bouzid, F. Charpillet, M.-Ch. Haton, B. Lâasri, H. Lâasri, P. Marquis, T. Mondot, and A. Napoli. *Le Raisonnement en Intelligence Artificielle*. InterEditions, Paris, France, 1991.

Janet Hitzeman and Massimo Poesio. Long distance pronominalisation and global focus. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics and* 36^{th} *annual conference of the Association for Computational Linguistics, Montréal, Canada*, pages 550–556, August 1998.

Xiaorong Huang. Planning reference choices for argumentative texts. In Robin Fawcett, Eduard H. Hovy, David McDonald, Marie Meteer, Donia Scott, and Koenraad deSmedt, editors, *Proceedings of the* 7^{th} *International Workshop on Natural Language Generation, Kennebunkport, Maine, USA*, pages 145–152, June 1994.

Proceedings of the 1^{st} *IEEE international symposium on Requirements Engineering, Los Alamitos, USA*, January 1993. IEEE, IEEE Computer Society Press.

Masato Ishizaki. *Mixed-Initiative Natural Language Dialogue with Variable Communicative Modes*. PhD thesis, University of Edinburgh, Scotland, 1997.

Joachim Karlsson. Software requirements prioritizing. In *Proceedings of the* 2^{nd} *IEEE International Conference on Requirements Engineering, Colorado-Springs, CO, USA*, pages 110–118, April 1996.

Richard Kittredge, Tanya Korelsky, and Owen Rambow. On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314, November 1991.

Robert Kowalski. *Logic for problem solving*. Elsevier Science Publishing Company, 1979.

Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

Renaud Lecœuche, Chris Mellish, Catherine Barry, and Dave Robertson. User-system dialogues and the notion of focus. *The Knowledge Engineering Review*, 4(13):381–408, 1998a.

Renaud Lecœuche, Chris Mellish, and Dave Robertson. A framework for requirements elicitation through mixed-initiative dialogue. In *Proceedings of the* 3^{rd} *International Conference on Requirements Engineering, Colorado-Springs, CO, USA*, pages 190–196. IEEE, IEEE Computer Society Press, April 1998b.

Renaud Lecœuche, Dave Robertson, and Catherine Barry. Using focus rules in requirements elicitation dialogues. In Thomas Dean, editor, *Proceedings of the* 16^{th} *International Joint Conference on Artificial Intelligence, Stockholm, Sweden*, pages 649–654. Morgan Kaufman, August 1999.

Benjamin Macias and Stephen Pulman. Natural language processing for requirements specifications. In Felix Remill and Tom Anderson, editors, *Safety-critical Systems*, chapter 4, pages 67–89. Chapman & Hall, 1993.

Kathleen R. McKeown. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41, 1985.

George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

Vibhu O. Mittal, Johanna D. Moore, Giuseppe Carenini, and Steven Roth. Describing complex charts in natural language: A caption generation system. *Computational Linguistics*, 24(3):431–467, 1998.

Jens-Uwe Moeller. Domain-related focus-shifting constraints in dialogues with knowledge based systems. In Giovanni Adorni and Michael Zock, editors, *Trends in natural language generation: An artificial intelligence perspective*, number 1036 in Lecture Notes in Artificial Intelligence, pages 188–204. Springer-Verlag, 1996.

Johanna D. Moore and Cécile L. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694, December 1993.

Ana M. Moreno. Object-oriented analysis from textual specifications. In *Proceedings of the* 9^{th} *international conference on Software Engineering and Knowledge Engineering, Madrid, Spain*, June 1997.

Sastry Nanduri and Spencer Rugaber. Requirements validation via automated language parsing. *Journal of Management Information Systems*, 12(2):9–19, 1996.

Miles Osborne and C K MacNish. Processing natural language software requirement specification. In *Proceedings of the* 2^{nd} *International Conference on Requirements Engineering, Colorado-Springs, Colorado, USA*, pages 229–236, 1996.

Derek Partridge. *Engineering Artificial Intelligence Software.* Intellect Books, Oxford, United Kingdom, 1992.

Justin Peterson, Kavi Mahesh, and Ashok Goel. Situating natural language understanding within experience-based design. *International Journal of Human-Computer Studies*, 41:881–913, 1994.

Livia Polanyi. A theory of discourse structure and discourse coherence. In W. H. Eilfort, P. D. Kroeber, and K.L Peterson, editors, *Papers from the general session of the* 21^{st} *regional meeting of the Chicago linguistics society, Chicago, Illinois, USA*, pages 306–322, April 1985.

Chris A. Reed and Derek P. Long. Content ordering in the generation of persuasive discourse. In Martha E. Pollack, editor, *Proceedings of the* 15^{th} *International Joint Conference on Artificial Intelligence, Nagoya, Japan*, pages 1022–1027. Morgan Kaufman, August 1997a.

Chris A. Reed and Derek P. Long. Ordering and focusing in an architecture for persuasive discourse planning. In W. Hoeppner, editor, *Proceedings of the* 6^{th} *European Workshop on Natural Language Generation, Duisburg, Germany*, March 1997b.

Rachel Reichman. *Getting Computers to Talk Like You and Me.* The MIT Press, Cambridge, MA, USA, 1985.

Ehud Reiter. NLG vs. templates. In *Proceedings of the* 5^{th} *European workshop on natural language generation, Leiden, The Netherlands*, May 1995.

Howard B. Reubenstein. Automated acquisition of evolving informal descriptions. Technical Report TR-1205, MIT Artificial Intelligence Laboratory, June 1990.

Colette Rolland and Camille Ben Achour. Guiding the construction of textual use case specifications. *Data and knowledge engineering journal*, 25(1–2): 125–160, March 1998.

Colette Rolland and C. Proix. A natural language approach for requirements engineering. In P. Loucopoulos and R. Zicari, editors, *Conceptual modelling, databases and CASE: An integrated view of the information systems development.* John Wiley & Sons, 1992.

Thomas L. Saaty. How to make a decision: The Analytic Hierarchy Process. *European Journal of Operational Research*, 48:9–26, 1990.

Roger C. Schank. Rules and topics in conversation. *Cognitive Science*, 1:421–441, 1977.

Samira Si-Saïd and Colette Rolland. Formalising guidance for the CREWS goal-scenario approach to requirements engineering. In *Proceedings of the* 8^{th} *European-Japanese Conference on Information Modelling and Knowledge Bases, Vamala, Finland*, May 1998.

Penelope Sibun. Generating text without trees. *Computational Intelligence*, 8 (1):102–122, 1992.

Teresa Sikorski and James F. Allen. A task-based evaluation of the TRAINS-95 dialogue system. In Elisabeth Maier, Marion Mast, and Susann LuperFoy, editors, *Proceedings of the ECAI Workshop on Dialogue Processing in Spoken Languages Systems, Budapest, Hungary*, volume 1236 of *LNAI*, pages 207–220, Berlin, August 1996. Springer. ISBN 3-540-63175-5.

Ian Sommerville. *Software Engineering*. Addison-Wesley, 5^{th} edition, 1995. ISBN 0-201-42765-6.

Ian Sommerville, Tom Rodden, Pete Sawyer, Richard Bentley, and Michael Twidale. Integrating ethnography into the requirements engineering process. In *Proceedings of the* 1^{st} *IEEE international symposium on Requirements Engineering, Los Alamitos, USA* IEE (1993), pages 165–173.

Ian Sommerville and Pete Sawyer. *Requirements Engineering: A good practice guide*. John Wiley & Sons, 1997. ISBN 0-471-97444-7.

Karen Sparck Jones and Julia R. Galliers. *Evaluating Natural Language Processing Systems*, volume 1083 of *Lecture Notes in Artificial Intelligence*. Springer, 1996.

Amanda J. Stent. Aspects of natural language generation. Technical Report 701, Computer Science Department, University of Rochester, November 1998.

Richard Thayer and Martin Dorfman, editors. *Software Requirements Engineering*. IEEE Computer Society Press, 2^{nd} edition, 1997.

Marilyn A. Walker. The effect of resource limits and task complexity on collaborative planning in dialogue. *Artificial Intelligence*, 85(1–2):181–243, 1996.

Marilyn A. Walker, Don Hindle, Jeanne Fromer, Giuseppe Di Fabbrizio, and Craig Mestel. Evaluating competing agent strategies for a voice email agent. In G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors, *Proceedings of the* 5^{th} *European Conference on Speech Technology and Communication, Rhodes, Greece*, pages 2219–2222. TYPOFFSET, Patras, Greece, September 1997a.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of the* 35^{th} *annual meeting of the Association for Computational Linguistics, Madrid, Spain*, July 1997b.

Paul T. Ward. The transformation schema: An extension of the data flow diagram to represent control and timing. *IEEE Transactions on Software Engineering*, 12(2):198–210, 1986.

Roel J. Wieringa. *Requirements Engineering – Frameworks for understanding*. John Wiley & Sons, 1996.