Using Focus Rules in Requirements Elicitation Dialogues

Renaud Lecœuche *† and **Dave Robertson**† and **Catherine Barry***

renaudl@dai.ed.ac.uk dr@dai.ed.ac.uk greboval@insa-rouen.fr

Abstract

Requirements engineering is a complex task which benefits from computer support. Despite the progress made in automatic reasoning on requirements, the tools supporting requirements elicitation remain difficult to use. In this paper we propose a novel approach where a tool's reasoning is intimately linked to the dialogue it has with its users. Because the dialogue is guided by rules ensuring coherence, the interaction with the tool is more natural. We discuss in detail the rules we use to organise the dialogue and how we apply them to the requirements elicitation tool. We present an evaluation of this approach demonstrating improvements in usability during the elicitation process.

1 Introduction

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 beneficial. This support requires formal specifications. However people are usually not trained to use formal specification languages. Task or domain specific languages smooth the learning curve to write 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. However, the interaction between the requirements engineer or future users and requirements elicitation tools has often been neglected. This is a problem since the easiest way to acquire the information for the requirements elicitation tools is often not the most acceptable in terms of human dialogue.

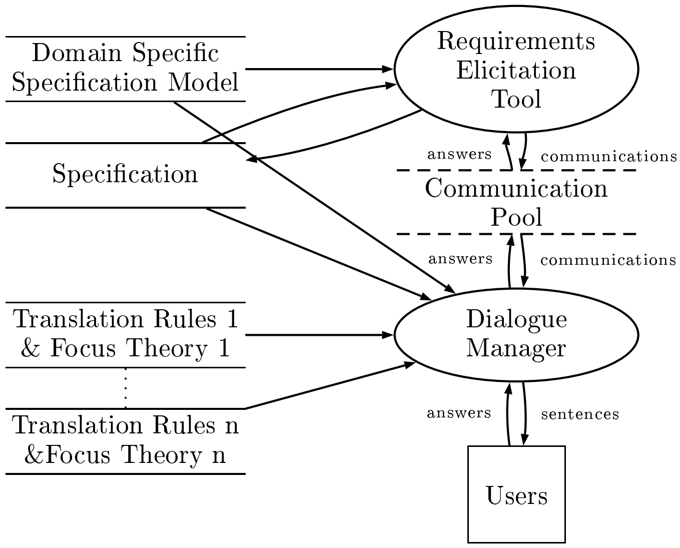

Figure 1: System data flow diagram

Adapting the dialogue so that users can make sense of the information provided by the tools is therefore important. On the other hand we do not want to force requirements elicitation tools to be aware of dialogue management strategies. A solution to this problem is to separate requirements elicitation tools and dialogue managers but make them interact and constrain each other.

We have developed a system which deals with this issue. This system communicates with its users to acquire requirements interactively. Its main feature is to organise the dialogue it has with users in order to ensure a coherent interaction. It is composed of two parts: a requirements elicitation tool and a dialogue manager as shown in figure 1 [Ward, 1986, details the graphical notation used].

The *Requirements Elicitation Tool* checks that the requirements entered by users are correct. It also provides guidance by making deductions on the requirements. It uses a *Domain Specific Specification Model* which contains the domain knowledge of the system about what

*PSI-LIRINSA, I.N.S.A., Place Emile Blondel, 76130 Mont-Saint-Aignan, France

†Department of Artificial Intelligence, 80 South Bridge, Edinburgh EH1 1HN, Scotland

can be elicited. It also accesses a *Specification* which represents the requirements that have been elicited so far. The specification is the instanciation of the domain specific specification model for the specific situation the users have described.

In our system, the requirements elicitation tool and the dialogue manager interact by means of "communications". The *Communication Pool* contains the communications to be output by the the dialogue manager and the answers provided by the users. The requirements elicitation tool puts communications in the communication pool depending on the deductions it makes and the dialogue manager puts users' answers to the communications as they become available.

The *Dialogue Manager* outputs the communications created by the requirements elicitation tool in natural language. It also enables users to write English sentences that are then transformed into a notation appropriate for the tool. It organises how the communications are output by means of *Focus Theories*. Focus can be defined informally as "the set of all the things to which participants in a dialogue are attending at a certain point in the dialogue and the point of view they have on these things" [Lecœuche *et al.*, 1998]. Focus theories ensure that the guidance provided by the tool is appropriate to the users' current task. The dialogue manager has access to different focus theories that can be "plugged-in" the system. Therefore we can adapt the level of text structure by using different focus theories. Focus theories seem well-suited for organising the kind of dialogues we consider here. Other techniques, such as those based on the intentions underlying the dialogue, would require the dialogue manager to know what the elicitation system is trying to achieve and what its plan is. For some elicitation systems, this knowledge may not be available. Focus theories require less information and allow a stronger separation between the elicitation system and the dialogue manager.

We have tested our system with two theories: a global focus theory organising the dialogue at a high level and a local focus theory (not presented here but based on centering [Grosz and Sidner, 1986]) organising the sentence-by-sentence flow of the dialogue. The dialogue organisation depends also on the communications available in the communication pool and therefore on the requirements elicitation tool. Conversely, the reasoning of this module is dependent on which communications the dialogue manager outputs since this directly influences users' answers. The focus theories have access, through the dialogue manager, to the specification and the specification model in order to re-use the knowledge represented in these data stores. The access to these datastores is mediated by *Translation Rules*.

In this paper, we present in section 2 one of the focus theories we use in our system. We advocate the use of a formal focus theory independent of any domain. Because the theory is formal, we can prove properties about the dialogues it allows. In section 3, we describe some translation rules we use to transform the domain knowledge used by our requirements elicitation tool into a form suitable to our formal focus theory. We show in section 4 an example of a requirements elicitation dialogue when using our dialogue manager. In that example, our system enables users to specify how to present a research group WWW site. We then summarise an evaluation of the effectiveness of this approach for requirements elicitation dialogues in section 5. The fact that focus rules improve the quality of texts is usually taken for granted and very few approaches provide a precise evaluation of their role. Finally we conclude the paper in section 6.

2 Focus Rules

In this section we present the formal focus theory that our system uses. The theory ensures that dialogues between the tool and its users are globally coherent. It is based on a simplification of Reichman's global focus theory [Reichman, 1985]. The basic idea is that the dialogue is supported by a sequence of changes to a focus spaces set. The focus space set contains focus spaces which represent the information in focus during part of the 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 some communications have been output 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 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.

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 at most controlled by one other space and a space cannot control a space already controlling it or controlling any of its controlling spaces.

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 the active focus space. They can also cause focus space activations to be modified and focus spaces to be created. In our theory, communications have one main subject which is a thing to be discussed, and possibly some other subjects which provide some context for the communication.

Our theory is composed of seven rules. The focus rules define the possible ways the dialogue may develop. Associated with an ordering, they represent what we expect to say next in the dialogue. The rules are based on four relations between the things that can be mentioned in the dialogue:

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.

Expression	Interpretation
$\mathcal{F}_i$	A focus space.
$active(\mathcal{F}_i)$	Focus space $\mathcal{F}_i$ is active.
$cont(\mathcal{F}_i, \mathcal{F}_j)$	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 to be output.
$subject(\mathcal{C}, X)$	X is a subject of communication $\mathcal{C}$.
$main(\mathcal{C}, X)$	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 in simple relation R with X_2.

Table 1: Notation used to formalise the global focus rules

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 already 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 already links it to the other.

The exact nature of these relations depends on the application domain. The translation from the domain relations to these relations is discussed in section 3.

Because focus rules only indicate what changes in the focus spaces (most things stay the same), we are faced with a simple version of the frame problem, i.e., how to indicate what changes and what does not. We tackle this problem by indicating the time during which a relation holds rather than asserting and deleting the relation itself [Kowalski and Sergot, 1986]. In particular, we represent the time in the dialogue as the sequence of communications and their associated focus moves 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 $\langle m_1(\mathcal{C}_1), m_2(\mathcal{C}_2), m_3(\mathcal{C}_3)\rangle$.

We define the $\succeq$ relation on time in the following way: $t_1 \succeq t_2 \Leftrightarrow \exists t_3.(t_1 = t_2 \cdot t_3)$ where $\cdot$ is the sequence concatenation operator. The $\succ$ relation 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 noted $start(P, t)$ where P is the relation and t the time. Relations that stop holding at a certain time are noted $end(P, t)$. 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

$$hold(P,t) \Leftrightarrow \exists t_s. \left(\begin{array}{l} t \succeq t_s \ \wedge\ start(P, t_s) \ \wedge \\ \neg\exists t_e. (t \succeq t_e \ \wedge\ t_e \succ t_s \ \wedge\ end(P, t_e)) \end{array} \right).$$

Because the notation used above is quite cumbersome, 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 ignore the time if it does not influence the truth value of the predicate.

The notation used to represent the rules is expressed in our simplified notation in table 1.

In the following rules, the time variable, t, and the communication variable, $\mathcal{C}$, are universally quantified. All other variables are assumed to be existentially quantified unless explicitly universally quantified. When a new space is created, it is assigned an unused number. This number is computed by the function $new(t)$ where t is the dialogue time from which the number of existing focus spaces can be deduced.

Rule 2.1 (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.*

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

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

Rule 2.2 (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 space. It may therefore serve to give background information on the things in the current active focus space.*

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

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

Rule 2.3 (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.*

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

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

Rule 2.4 (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.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ main(\mathcal{C}, X_2) \wedge \\ sim_t(R, X_1, X_2) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new(t')}), t') \\ \wedge\ \forall X.(X \in_t \mathcal{F}_i \\ \quad \rightarrow start(X \in \mathcal{F}_{new(t')}, t')) \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \quad \rightarrow start(X \in \mathcal{F}_{new(t')}, t')) \\ \wedge\ cont_t(\mathcal{F}_j, \mathcal{F}_i) \\ \rightarrow start(cont(\mathcal{F}_j, \mathcal{F}_{new(t')}), t') \end{pmatrix}$$

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

Rule 2.5 (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 does not output any communications. It is always used in conjunction with one of the other rules which outputs a communication.*

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

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

Rule 2.6 (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.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ main(\mathcal{C}, X_1) \wedge \\ \forall j.(X_1 \notin_t \mathcal{F}_j) \wedge \\ \forall R.(\neg\exists X.(X \in_t \mathcal{F}_i \wedge \\ (dir_t(R, X, X_1) \\ \vee spe_t(R, X, X_1) \\ \vee gen_t(R, X, X_1) \\ \vee sim_t(R, X, X_1)))) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(\\ \quad active(\mathcal{F}_{new(t')}), t') \\ \wedge\ start(\\ \quad cont(\mathcal{F}_i, \mathcal{F}_{new(t')}), t') \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \rightarrow start(X \in \mathcal{F}_{new(t')}, t')) \end{pmatrix}$$

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

Rule 2.7 (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.*

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

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

The dialogue starting state for this theory is $active_{\langle\rangle}(\mathcal{F}_1)$ with $\mathcal{F}_1 =_{\langle\rangle} \{\}$.

Now that the rules have been formalised, it is possible to prove properties of them. For example, we can verify that in any dialogue, there is exactly one active space. This can be shown by induction on the dialogue time. At time $t = \langle\rangle$, the property is verified since there is only one focus space and this space is active. If we then suppose that for any time $t \prec T$, the property holds, it also holds at time $t = T$ because (1) every move starting a new active relation ends the current active relation, (2) no move ends the current active relation without starting a new one. Checking these properties is difficult in most other focus theories because they are not formalised.

2.1 Rule ordering

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. 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. If such a communication does not exist, we 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 communication to output is already a member of a space controlling the current active space then a pop is the preferred move. This avoids reintroducing concepts that we are expecting to return to later in the dialogue.

3 From Domain Knowledge to Focus Rule Representation

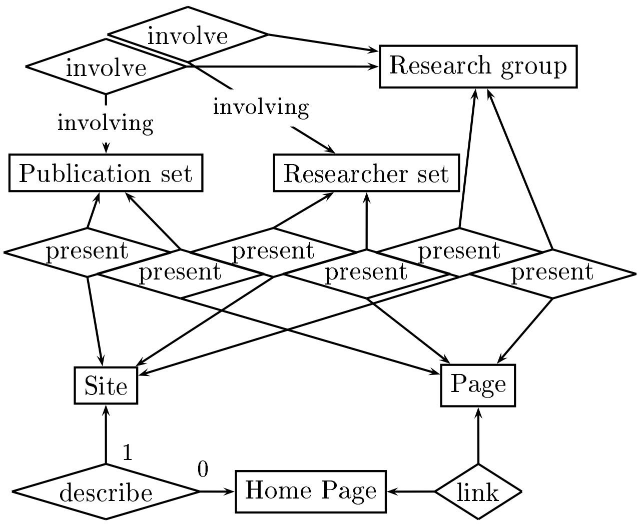

Figure 2: Specification model example

Because the focus rules we presented in section 2 are based on generic concepts such as direct, specialisation, generalisation or simple relations which are not necessarily used in the application domain of the system, we need to bridge the gap between the representation used in this domain and the representation used by the focus rules. This is done by "translation rules". These rules allow the use of this focus theory in different domains and for different applications.

We have created a set of translation rules to map the specification model used by our requirements elicitation tool to the generic concepts used by our formalised focus theory. In our system, the specification model is represented by an Entity-Relationship (ER) model [Chen, 1976]. A part of the model used to check the specification of a WWW site describing a research group is presented in figure 2 [Wieringa, 1996, details the graphical notation used]. A research group involves a set of researchers and a set of publications. These three entities can be presented by a single WWW page or by a WWW site, i.e., a group of pages. Sites can be described by a home page. Pages and home pages can be linked together.

The main idea behind the translation rules is to allow things that are related to the current focus and cannot be accessed easily otherwise to be mentioned immediately. We only present here two examples of such rules (These rules can be formalised [Lecœuche, 1999]. We do not show the formalisation here as it would require us to also explain the formal framework used to represent the specification model):

- An entity is in direct relation with another if it may not exist at all and if it cannot exist without the other one. In our example, a home page is an optional entity whose existence requires the existence of a site. There is therefore a direct relation from the site to the home page.
- If an entity can be linked to several others playing the same role, then there is a relation of specialisation from this entity to the others. In our example, the research group involves the researcher set and the publication set which fill the same role `involving`. These two elements are therefore in a specialisation relation with the research group.

Using the translation rules, the ER relations presented in figure 2 are associated with the following focus relations:

Direct relations: `present` from publication set, researcher set and research group to site and page, `describe` from site to home page.

Specialisation relations: `involve` from research group to publication set and researcher set, `link` from home page to page and conversely.

Generalisation relations: `involve` from publication set and researcher set to research group, `present` from site and page to publication set, researcher set and research group.

Simple relations: All other relations.

Basing the global focus relations on the specification model is a great advantage since it enables us to re-use the domain knowledge of the tool checking the specification. Moreover the translation rules we presented here are dependent on the ER model structure but not on the application domain itself. This makes them more generic. As a consequence we have been able to use them in other domains than the WWW site design one. In particular we have applied our system for a simple library loan process description tool and for a more generic ER model acquisition tool. The translation rules could however be made more domain dependent if need be.

4 Example

In this section we present an example of a requirements elicitation dialogue. Our system enables users to specify how to present a research group by a WWW site. Once this has been done, the specification can be transformed into HTML and the site can be displayed by a WWW browser. Our system allows users' intervention in the elicitation process but we show here an example where it has the initiative and is in charge of acquiring the requirements.

The system uses templates to generate sentences and interpret users' answers. Depending on the focus information available, the templates can be adapted to produce pronouns, or definite noun phrases. Entities introduced in the dialogue can be referred to using a unique identifier provided in the dialogue by the system.

Using the results of the translation rules, the system tries to keep elements in direct relation together as explained in section 2.1. This is shown in the example

S:	Research group #rg1 is the research group. Do you want a site presenting it?
U:	Yes
S:	A site presents the research group. Which site presents the research group?
U:	A new site.
S:	Site #s1 presents the research group. Do you want a home page describing site #s1?
U:	Site #s1 is described by a new home page.
S:	Home page #hp1 describes site #s1. What is the title of home page #hp1?
U:	"Research group".
S:	Do you want a page linked from it?
U:	I want to speak about the researcher set
S:	OK. Researcher set #rs1 is the researcher set. It is involved by the research group. Do you want a page presenting it?

Table 2: Dialogue example

presented in table 2. The system first deals with everything directly related to the research group, i.e., site and home page, in the same focus space, $\mathcal{F}_1$. Then it selects a specialisation relation from the home page to a page. This move opens a new focus space, $\mathcal{F}_2$, under the control of the previous one. However, the user redirects the dialogue by choosing another specialisation relation from research group to researcher set. This move closes the controlled focus space and opens another one, $\mathcal{F}_3$, under the control of the first focus space. The final state of the dialogue is presented graphically in figure 3. This dialogue is more coherent than a dialogue where the system is free to choose the topic and could possibly shift randomly between research group, home page, site and researcher set.

5 Evaluation

A detailed evaluation of our focus theories has been carried out [Lecœuche, 1999]. In this section we summarise the results of initial tests with users.

We tested our system by asking people to read transcripts of requirements elicitation dialogues generated with and without focus theories and asking them to compare the dialogues pair-wise for coherence [Saaty, 1990]. Transformations can then be made on these comparisons to compute the overall ranking of the theories. For example, a theory with a value of 60 is considered three

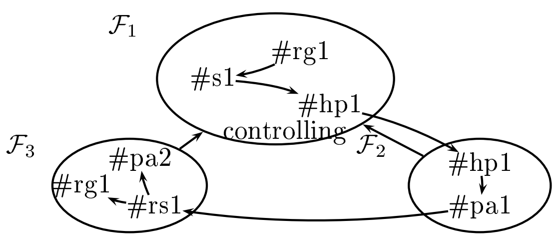

Figure 3: Dialogue evolution

Local and global theories order	39
Global theory order	34
Elicitation module order	17
Local theory order	10
	100

Table 3: Evaluation

times more coherent than a theory with a value of 20. The values of all the theories add up to 100.

The experiment involved six persons for 90 minutes. Six dialogues were evaluated. Each participant compared four dialogues, resulting in 36 pair-wise comparisons. The results are presented in table 3. (The consistency ratio is equal to 0.0993 which indicates reliable results. The participants may however not be representative of the whole user community.) They show that our global focus theory was judged to improve the global coherence of the dialogue compared to following the requirements elicitation tool order, i.e., without using focus rules. The results also show that the local theory we use in our system performs badly on its own. This may be explained by the tendency of local theories to produce "spaghetti" dialogues without global structure. The local and global theories together perform somewhat better than the global theory alone. This can be explained by the local theory improving the transitions between communications in each global focus spaces and the transition between global focus spaces themselves.

This evaluation shows that our approach provides improvements in perceived dialogue quality. However using any individual focus theory may not be sufficient to achieve this result as demonstrated by the poor evaluation of the local focus theory on its own.

6 Conclusion

We have presented an interactive system for requirements elicitation dialogue. Our system uses focus rules to improve the quality of its interaction with users. Focus rules have rarely been used in interactive systems, especially to direct the system's reasoning. Other systems used for requirements elicitation [Rolland and Ben Achour, 1998; Reubenstein, 1990] do not take them into account and leave users in charge of picking up the relevant information from the system output. We have discussed a formal focus theory. Because of its formality, properties of the theory can be proven mathematically. We then presented "translation" rules, bridging the gap between the representation used by the focus theory and the representation used for the domain knowledge. We then presented an example of dialogue showing the influence of the focus theory. Finally, we showed that our approach was judged to improve the perceived quality of requirements elicitation dialogues. Although we presented our results in the domain of requirements engineering, we believe that our theory could be used in other domains where the quality of interaction with users is important. We now plan to investigate how the theory presented in this paper could be integrated with other techniques (such as intentional ones in case the elicitation system is able to indicate its plan) and to use it in multi-agent systems where one of the agent would be in charge of the natural language interation with users.

References

[Chen, 1976] 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.

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

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

[Lecœuche *et al.*, 1998] 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, 1998.

[Lecœuche, 1999] Renaud Lecœuche. *Formalisation and Evaluation of Focus Theories for Requirements Elicitation Dialogues in Natural Language*. PhD thesis, University of Edinburgh, 1999.

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

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

[Rolland and Ben Achour, 1998] 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.

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

[Ward, 1986] 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.

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