A Framework for Requirements Elicitation through Mixed-Initiative Dialogue

Renaud Lecœuche *† renaudl@dai.ed.ac.uk
Chris Mellish † chrism@dai.ed.ac.uk
Dave Robertson † dr@dai.ed.ac.uk

* PSI-LIRINSA
Université de Rouen
Mont-Saint-Aignan, France, 76130

† Department of Artificial Intelligence
University of Edinburgh
Edinburgh, Scotland, EH1 1HN

Abstract

In this paper we present our work on requirements elicitation. The elicitation process is a complex task which necessitates computer support. Elicitation systems should ideally help their users check the correctness of the specifications obtained but also actively guide them in the acquisition of the requirements. We consider hereafter systems that communicate in natural language. We describe a framework that tries to improve the quality of the guidance it provides to its users by taking into account natural language constraints. We discuss the need for a theory of natural language dialogue structure, and we show how we have integrated such a theory within an early prototype of an elicitation system.

Keywords: requirements elicitation, guidance, knowledge-based systems, natural language, mixed-initiative, WWW

1 Introduction

Numerous frameworks, such as goal-oriented strategies [3], scenario-based strategies [12] and rapid prototyping [1, 19, 8], have been suggested as aids to the acquisition of software requirements. Much less attention has been paid to the control of dialogue taking place between the users and the system whilst using such frameworks [2]. Our ability to develop sophisticated formal frameworks makes this deficiency more acutely felt, since increases in formality are often accompanied by greater difficulty in understanding and using the frameworks [22]. Frameworks for requirements acquisition are not normally accompanied by theories of the types of dialogue which they support. However, some theories of dialogue structure are beginning to emerge, notably from the natural language processing community. It is interesting to examine how the structure of requirements acquisition frameworks, which attempt to control the development of a requirements specification, may be linked to the structure of theories of dialogue, which attempt to control dialogue according to the conventions of discourse.

Many approaches to the problem of natural language interaction for requirements elicitation have considered it as a translation problem: the users write specifications in natural language and the system expresses them in a formal framework [6, 7, 15, 16, 17, 26]. As the natural language statements are directly and automatically translated into specifications that should ideally be consistent, complete and coherent[1], they should themselves have those properties. If there are conflicts in the natural language requirements, however, these may not be resolved because the system has no way of knowing what the user really meant. As a result, the task of the user remains difficult. Although the formal notation of the target framework is hidden by the natural language, its constraints are still fully present and the user must consider them to write correct specifications.

In order to actively help users in the process of writing the requirements, the elicitation system must interact with them. The emphasis here is no longer on translating requirements but on actively extracting them through dialogue with the users. This task is complicated by the fact that there are numerous ways of carrying out the requirements elicitation. In other words the requirement frameworks alone are not providing enough constraints to ensure a unique elicitation process. An arbitrary choice could be made, but forcing the users to adopt a predefined method is usually not possible as it would make the elicitation process very difficult to follow and understand. The system must therefore be able to adapt itself to various elicitation

[1] The properties that requirements should exhibit are still a matter of debate [13]. The ones we cite are just used as examples. The point is that, whatever the properties, the specifications are constrained.

methods. On the other hand it is necessary for the system to make choices in order to provide active guidance. A "least-commitment" strategy, such as asking the user at every choice point what to do next, is not a useful approach [5]. One way of offering guidance without restricting the users too much is by using natural language constraints (NLC) and mixed-initiative. NLCs ensure that the system adopts a strategy that will guide the users in a natural and understandable manner. Mixed-initiative ensures that the users can redirect the process and are not limited to a single elicitation method. Using such an approach reduces some of the problems discussed above. The specifications do not need to be immediately correct as they will be checked and reworked by the system. The formal framework is hidden from the user but is still there to ensure the correctness of the specifications. Guidance is continuously offered through dialogue, which is influenced by but does not directly follow the steps of construction of the specification. Moreover, because of the mixed-initiative, the user is still free to divert the course of dialogue by volunteering strategic information.

The integration of NLCs with requirements elicitation rules requires two major steps. The first step consists in the definition of the NLCs. NLCs related to mixed-initiative dialogue have been studied by the natural language processing community since the '70s. Different theories have been proposed. We will present them and explain how they can direct the dialogue between the users and the elicitation system. The second step consists in the actual integration of the NLCs in the requirements elicitation process. The integration of these constraints with the requirements elicitation constraints is a recent domain of research. We will present in detail our way of integrating the NLCs with the requirements elicitation process.

This paper is organised as follows. In section 2, we present some current dialogue theories and discuss how they can influence the dialogue between the users and the elicitation system. In section 3 we describe our system and how we have integrated the NLCs with the elicitation process. We then show in section 4 how the system works on some examples. In section 5 we compare our system with related work. Finally we conclude in section 6 with some shortcomings of our approach and future work.

2 Dialogue theory

To study the structure of dialogues during the elicitation task we have recorded several dialogues in the domain of research group WWW site construction. The participants in the dialogue are a WWW site expert acting as the elicitation system and a research group member acting as the user of the system. The aim of the dialogue is to elicit the structure of the WWW site the user wants to create. By structure, we mean the organisation of the site and the navigation constraints. This corresponds to the third part of the usual decomposition of WWW sites in content, i.e., what is the site about, presentation, i.e., how is the information presented, and navigation, i.e., how users move between pages.

This study has provided two interesting results:

- There are numerous ways of describing the requirements. Although the information conveyed in the different dialogues was basically the same the way it was elicited varied a lot. Every dialogue is organised differently from the others. This confirms the fact that numerous elicitation methods are possible.

- There are constraints on the structure of the dialogues. Although the dialogues were very different from one another, each of them presents a clear internal structure.

A system constraining the elicitation process to a small set of predefined paths or a system oblivious of the dialogue structure would make the elicitation process difficult for humans to follow. It is therefore necessary to take into account rules explaining dialogue structure. Focus rules are such rules.

Most focus theories are based on the notion of focus space. A focus space contains what is important at a certain point in a dialogue, i.e., the things to which the participants in a dialogue are attending. As the dialogue evolves, new focus spaces are created when new things are discussed and then closed; or old focus spaces are reopened when old things are re-introduced. Focus spaces are organised in a stack [9] or a tree [14] or a more complex structure [20] depending on the theory.

The important point of these theories is that the evolution of the dialogue, i.e., the shifts between the different focus spaces, is constrained. For example, focus shifts can be constrained by the task to be achieved by the dialogue participants [9], the participants' intentions [10], or the domain structure [14]. By using these dialogue constraints we can define elicitation paths that are easily understood by the users without restraining them to predefined paths.

There are a number of issues that we have to deal with and that are not clearly tackled by the focus theories:

- Re-opening closed focus spaces is not clearly explained. We need however such rules if the users realize that some requirements are incorrect and should be reworked.

- Linguistics phenomena marking the focus shifts in the dialogue, such as cue words, are different from one theory to the other. However, without them the dialogue would be difficult for users to understand.

- The shifts in focus also depend on domain properties. For example, the shift to a link between pages can only be done if there are at least two pages that already have been considered.

We propose some solutions to these problems:

- We have incorporated domain specific shift rules in our system[2]. As will be explained in section 3, this is achieved by mapping shift rules with domain specific elicitation rules. The resulting rule that will effectively be applied combines the two separate rules and requires the two set of constraints to be satisfied.

- The cue words used in the dialogues we have collected are different from one dialogue to the other. Therefore it seems more practical to use a standard cue word rather than debating on the best possible cue word for each particular shift. More complex combinations will certainly be necessary to provide more natural sounding dialogues.

- Rules for determining backward focus shifts, i.e., shifts to old focus spaces, are still an open problem. If a concept is introduced only once in the dialogue, a backward shift to that concept is easy to do: we just need to put it back into focus with its associated focus space. However, if a concept is introduced more than once, choosing the right shift remains an open question. In our system, the shift to a previous focus is supported by jumping back to its latest occurrence in the dialogue. While certainly not fully satisfactory, this approach performs reasonably well.

3 System

A major point of this paper is to show how the integration of NLCs with a usual requirements elicitation system can be achieved. We especially tried to minimise the impact of adding the NLCs on the elicitation system. As a result our system can run with or without NLCs. This enables us to develop semi-independently the set of elicitation rules and the set of focus rules. This feature should also help us evaluate the usefulness of the NLCs by turning on and off the NLCs and comparing the resulting dialogues.

The general architecture of our system is described in figure 1.

The system is composed of 5 main components:

Specification model describes the entities that should be elicited and their relations. It is currently based on a domain-specific object-oriented framework: the specification model is composed of classes such as site, sub-site, page, and of relations such as part-of or link. Classes and relations can have attributes. However the framework could be replaced by an existing requirements representation one such as KAOS [3, 4]. Elicited specifications which are instances of the classes defined in the model, are incrementally added to the model as the elicitation process progresses.

Dialogue model describes the entities that can be spoken about during the dialogue and their relations. It is currently based on an object-oriented framework[3]. Dialogue entities, i.e., instances of the concepts and links defined in the model, are added to the model as the dialogue takes place. The dialogue model also contains the focus spaces.

Elicitation rules define how particular pieces of information are acquired. They are based on a precondition-action template. The preconditions can test for specific structures in the specification model.

Dialogue rules define how concepts are created and how the focus can evolve. They are based on a precondition-action template. The preconditions can test for specific structures in the dialogue model.

Mapping rules define how the concept and link instances used in the dialogue are related to the entities manipulated by the elicitation system. The mapping is done between a subset of the specification model and a subset of the dialogue model. Therefore some entities, part of the *unmapped region* of the specification model, may not be considered in the dialogue. This allows a form of implicit requirements [11] where the system takes in charge the creation of specifications hidden from the user. Concepts that are not directly used in the specifications can also be spoken about. These concepts form the *unmapped region* of the dialogue model.

The mapping rules define a relation of compatibility between the elicitation rules and the dialogue rules. An elicitation rule and a dialogue rule are compatible if and only if:

- the rules act locally on the unmapped region of the specification model and on the unmapped region of

[2] Our system is a very early prototype working in the domain of WWW site elicitation. Using the system the users can specify the site organisation and navigation. Our current system contains a little more than 10 focus rules for around 15 concepts and 15 attributes in the knowledge base. HTML pages can be automatically generated from the specifications.

[3] In order to distinguish the dialogue model from the specification model, a class in the dialogue model is called a concept and a relation a link.

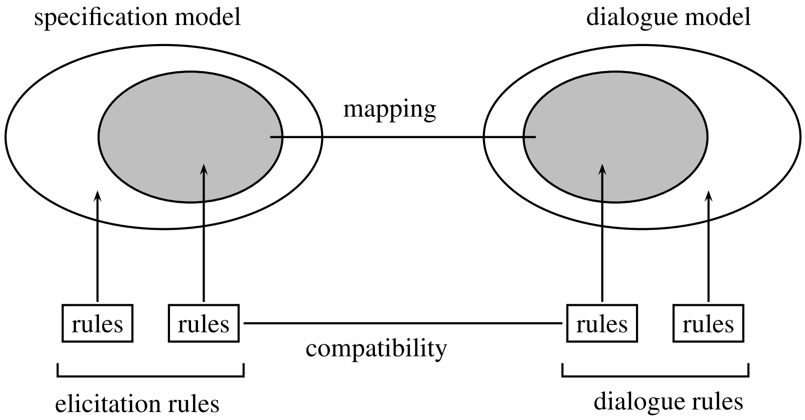

Figure 1: System architecture

the dialogue model respectively. These rules represent a special case of compatibility that do not depend on the mapping. They are represented in separate boxes in figure 1.

- the elicitation rule creates a class or relation instance that is mapped to a concept or link instance created by the dialogue rule and conversely.

- the elicitation rule sets the value of an attribute of a specification instance that is mapped to a dialogue instance attribute whose value is set by the dialogue rules and the two values are compatible. We do not detail here the compatibility rules between values. They just ensure that the specification and dialogue attributes have related values.

The compatibility rules enable the system to synchronise the requirements elicitation process with the dialogue. The specifications and dialogue rules are written independently but the mapping between the specification and dialogue models ensure that they are executed coherently. The compatibility rules also enable the system to reduce the number of questions asked by using the mapping between specifications and dialogue entities to avoid redundant questions.

Checking the compatibility of the rules ensures that they abide by the domain specific elicitation constraints as well as the dialogue constraints. It also avoids putting any overhead on the user as similar questions are eliminated.

4 Examples

In this section we present examples of the dialogue phenomena our system can handle and how they influence the elicitation process. The system tries to elicit the structure of a WWW site. In other words it tries to know what are the pages making the site and how they are linked.

No attention has been paid yet to the presentation of the system utterances. We are mainly interested in the structure of the dialogue and the capability of the system to direct or reset the focus as the user wants. User's answers are also very restricted in our current system. We plan to extend the possible answers with restricted natural language undestanding by using methods based on the focus mechanism [21]. We describe here three dialogue phenomena:

Example 1 *Ignoring questions*

> Let's talk about the site front_page
> —: ok.
> What is the topic of the page ?
> —: ignore.

`ignore` is a keyword indicating that the user does not know the answer to that question at that moment. The system tries to elicit a maximum of information before shifting the focus of dialogue, thus minimising the complexity of the dialogue. However, the users can indicate that they would rather come back to a question later by using the `ignore` keyword. The system then searches for another question which may involve changing the dialogue focus.

Example 2 *Resetting the focus*

> Which topic do you want to discuss ?
> —: front_page.
> What is the topic of the page ?
> —: research_group.

The system is usually in charge of setting and resetting the focus appropriately as the dialogue progresses.

However, if the users disagree with the automatic setting of the focus, they can indicate it. The system will then ask for the topic the users want to discuss and will reset the focus accordingly. The elicitation process restarts in that new situation.

These two phenomena give mixed-initiative to the users in the elicitation process: they can ignore questions and redirect the process.

Example 3 *Using information implicit in the dialogue structure*

> Let's talk about the navigation
> —: ok.
> Which page do you want to consider ?
> —: current_publications.
> Do you want to create a new internal_link ?
> —: yes.
> What is the destination of that link ?
> —: past_publications.

An `internal link` links two pages of the site. Its anchor is the page actually in focus while its destination is the page indicated in this question. Putting the anchor page in focus before doing the elicitation enables the system not to ask what the anchor of each link is. It is by convention the page focused on. Therefore, the information that is implicit in the focus makes the elicitation process less burdensome.

The number of dialogue rules needed to allow the three phenomena described is small (around 10). However handling these phenomena also requires that the elicitation system can redirect its process to the focused part of the specifications. This may limit the re-use of existing elicitation systems.

Once the requirements for the site have been elicited the system can create actual WWW pages using presentational rules. These rules define the lay-out of the pages based on the elicited information[4]. Using our system we were able to create a small WWW site (composed of 7 pages grouped in 3 subsites and linked by 27 links) with a simple dialogue (35 exchanges including 1 user-initiated focus redirection).

5 Related work

In this section we compare our approach with two pieces of related work.

The requirements engineering process has been extensively studied in the F3 project [18, 23, 25]. In that project requirements elicitation is seen as a spiral. One turn of the spiral represents the reasoning in a *localised* context to solve one identified problem. Links between turns represent shifting from one problm to another through context dependency. Rolland defines context dependency as follows [24]:

> "Context-dependency is the key element of our global process approach. In order to guide the requirements engineer in his progression along the spiral we propose to predefine dependencies."

Our approach is similar to the F3 one in that context dependencies are guiding the reasoning of the system. This approach differs from ours on two major aspects:

- The context-dependencies in F3 are predefined. Therefore the users are forced into a particular method for eliciting the requirements[5]. For example, it is not clear how the users can switch back to a previous topic if they want to.
- The context-dependencies are based on elicitation process properties. NLCs are not taken into account. While being a huge improvement on unguided elicitation approaches, the F3 system may still be difficult to understand for its users since it does not include NLCs.

Another piece of work related to our approach is on topic explanation [27]. In this work, the rules used by an expert system are decomposed into topics.

Each topic is characterised by a landmark and "has a description, consisting of background information necessary or useful to someone wanting to understand that topic."

The system uses these landmarks and the associated descriptions to explain its reasoning to its users. Each time a landmark is encountered during the reasoning, the system stops and explains what is going to be worked on. Topics and landmarks are therefore the main components in the dialogue between the system and its users. The systems also presents with the topic description all the methods it knows to solve the problem at hand. The users can then choose the method they desire. A certain mixed-initiative is thus built in the system.

Our approach differs from the topic explanation one on several accounts:

- We use the notion of topic not only to explain the reasoning of the system but also to guide it. Topics directly influence the firing of focus rules that in turn modify the applicability of the elicitation rules.

[4] Our research group WWW site is currently generated by this sort of rule. The content of the site is stored in the form of prolog terms that are interpreted by the rules to create the HTML pages.

[5] In fact, there may be choice points where the users can choose between competing paths. However the choices are limited.

- We separate the definition of the topics from the system rule base using a mapping between the two. This enables us to introduce topics that have no equivalent in the system reasoning.
- The mixed-initiative that we allow in our system is more important than just a choice of methods. The users can switch back to entirely different topics.

In spite of those differences, the idea behind our system and that behind topic explanation are very similar. The mapping between the dialogue model and the specification model creates an implicit division of the latter into topics.

Our system improves the two previous approaches by allowing a dialogue based on NLCs and by allowing more freedom in the mixed-initiative.

6 Conclusion and Future Work

In this paper we have presented our framework for requirements elicitation through mixed-initiative dialogue. We first described what the current theories of dialogue are and how we can adapt them for requirements elicitation. We then showed how those theories can be implemented to cooperate with the elicitation rules. We finally compared our approach with some related work.

There are a number of remaining problems:

- The dialogue theory we use is incomplete. The cases of complex focus shifts are not adequately handled. Our focus rules are also domain specific and lack generality. We plan to remedy these two problems by studying in greater depth the dialogues we have collected.
- The system does not have a proper user interface. We need to interface our system with a natural language generation tool. Given the fact that the NLCs are separated from the elicitation process, this should be relatively easy. We are planning to see how we can link our system with existing upper-models, thus facilitating natural language production.
- Since the system is a very early prototype, no tests have been done on its usability for casual users. We plan to study the impact of the NLCs on the quality of dialogue using the capability of the system to use or not use the focus rules to guide the elicitation process.

References

[1] S. Andriole. Fast, cheap requirements: Prototype, or else! *IEEE Software*, 11(2):85–87, March 1994.

[2] Janis Bubenko, Colette Rolland, P. Loucopoulos, and V. DeAntonellis. Facilitating fuzzy to formal requirements modelling. In *Proceedings of the* 1^{st} *IEEE International Conference on Requirements Engineering, Colorado-Springs, USA*, April 1994.

[3] Anne Dardenne, Axel van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20:3–50, 1993.

[4] Robert Darimont. *Process Support for Requirements Elicitation*. PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium, June 1995.

[5] George Ferguson, James F. Allen, and Brad Miller. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proceedings of the* 3^{rd} *conference on Artificial Intelligence Planning Systems, Edinburgh, Scotland*, pages 70–77, May 1996.

[6] Norbert E. Fuchs and Rolf Schwitter. Attempto - controlled natural language for requirements specifications. In *Proceedings of the* 7^{th} *ILPS workshop on logic programming environments, Portland, Oregon, USA*, 1995.

[7] Norbert E. Fuchs and Rolf Schwitter. Attempto controlled English (ACE). In *Proceedings of the* 1^{st} *international Workshop on Controlled Language Applications, Leuven, Belgium*, March 1996.

[8] Hassan Gomaa. The impact of prototyping on software system engineering. In Richard Thayer and Martin Dorfman, editors, *Software Requirements Engineering*, pages 431–440. IEEE Computer Society Press, 2^{nd} edition, 1997.

[9] Barbara J. Grosz. The representation and use of focus in dialogue understanding. Technical Report 151, SRI International, July 1977.

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

[11] Jeff Hagopian and Theresa Maxwell. Explicit and implicit resources: A simplified approach to user requirements modeling. In *Electronic Proceedings of the Space Ops 96 Symposium (http://www.op.dlr.de/SpaceOps/spops96/misplan/track3.htm)*, 1996.

[12] Pei Hsia, Jayarajan Samuel, Jerry Gao, David Kung, Yasufumi Toyoshima, and Chris Chen. Formal approach to scenario analysis. *IEEE Software*, 11(2):33–41, March 1994.

[13] O. Lindland, G. Sindre, and A. Solvberg. Understanding quality in conceptual modeling. *IEEE Software*, 11(2):42–49, March 1994.

[14] Kathleen F. McCoy and Jeannette Cheng. Focus of attention: Constraining what can be said next. In Cécile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, chapter 4, pages 103–124. Kluwer Academic Publishers, 1991.

[15] Luisa Mich. NL-OOPS: From natural language to object oriented requirements using the natural language processing system LOLITA. *Natural Language Engineering*, 2(2):161–187, 1996.

[16] 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.

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

[18] Véronique Plihon and Colette Rolland. Modelling ways-of-working. In *Proceedings of the* 7^{th} *international Conference on Advanced information Systems Engineering, Jyvaeskylae, Finland*, June 1995.

[19] Balasubramaniam Ramesh and Luqi. Process knowledge based rapid prototyping for requirements engineering. In *Proceedings of the* 1^{st} *IEEE international symposium on Requirements Engineering, Los Alamitos, USA*, pages 248–255. IEEE, IEEE Computer Society Press, January 1993.

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

[21] Charles Rich and Candace L. Sidner. Collagen: When agents collaborate with people. In *Proceedings of the* 1^{st} *Conference on Autonomous Agents, Marina del Rey, California, USA*, February 1997.

[22] David Stuart Robertson, Mike Uschold, Alan Bundy, and Robert Muetzelfeldt. The ECO program construction system: Ways of increasing its representational power and their effects on the user interface. *International Journal of Man Machine Studies*, 31(1):1–26, 1989.

[23] Colette Rolland. Modeling the requirements engineering process. In *Proceedings of the* 3^{rd} *European-Japanese Seminar on Information Modelling and Knowledge Bases, Budapest, Hungary*, June 1993.

[24] Colette Rolland. A contextual approach for the requirements engineering process. In *Proceedings of the* 6^{th} *international conference on Software Engineering and Knowledge Engineering, Vilnius, Lithuenia*, September 1994.

[25] Colette Rolland, Carine Souveyet, and Mario Moreno. An approach for defining ways-of-working. *Information Systems Journal*, 20(4), 1995.

[26] Rolf Schwitter and Norbert Fuchs. Attempto - from specifications in controlled natural language towards executable specifications. In *Proceedings of the GI EMISA Workshop, Tutzing, Germany*, 1996.

[27] Richard W. Southwick. Topic explanation in expert systems. In B. Kelly and A. Rector, editors, *Research and Development in Expert Systems V*, pages 47–57. Cambridge University Press, 1989.