CREWS Report Series 97-09

Linguistic Instruments for the Integration of Scenarios in Requirement Engineering

(Position Paper)

Camille Ben Achour
camille@univ-paris1.fr

Proceedings of the Third International Workshop
on Requirements Engineering:
Foundation for Software Quality (REFSQ'97),
Barcelona, Catalonia, Tunis, June 16-17, 1997

Linguistic Instruments for the Integration of Scenarios in Requirement Engineering[1]

(Position Paper)

Camille Ben Achour

camille@univ-paris1.fr

1. Introduction

An increasingly popular approach for performing Requirement Engineering (RE) is scenario based design. Scenarios, examples, mock-ups, and prototypes have also attracted considerable attention in Human Computer Interaction (HCI), Software Engineering (SE), and Information System (IS) communities. Loosely all these ideas can be called scenario based design. A close survey of the literature on existing scenario approaches [Rolland et al 97] [Ben Achour and Ralyte 97] showed that scenarios are proposed in a large variety of forms and usages. However there is a marking trend to define scenarios as textual descriptions of external views of the designed system behaviour. From the linguistic point of view, one could identify scenarios to stories composed of actions completed together with informations of static nature. In the HCI community scenarios have been proposed as detailed descriptions of a usage context so design decisions can be reasoned about [Carroll 95]. They are also small scale examples of existing products used to anchor discussion about different design theories [Young et al.87]. In SE "use cases" have been developed as informal narrative descriptions of use, responsibilities and services within object oriented design [Jacobson et al 92], [Jacobson 95]. Scenarios in the IS community have evolved to the concept of a rich picture which gives the social and environmental settings of a required system so that arguments can be developed about the impact of introducing technology, and the matching between user requirements and task support provided by the system [Kyng 95]. Finally in RE, scenario scripts have been proposed as test data for checking dependencies between a requirements specification and the environment in which it has to function [Potts et al 94] [Holbrook 90].

If they describe how to use scenarios for the elicitation, exploration, or validation of requirements, very few approaches say how scenarios could be integrated into different existing methods and supported by tools. Yet, the industrial practitioners see methods as composite molecular artefacts, made of chunks of several different methods, and to which emerging techniques, like the scenario approach, might be integrated. There is

[1] This work is partly funded by the Basic Reserch Action CREWS (ESPRIT N° 21.903). CREWS stands for Cooperative Requirements Engineering With Scenarios.

thus a strong need to define supports for the integration of scenarios into different methods. If many linguistic based instruments have already proposed solutions to identify requirements diagrams out of textual requirements, few envisage to take scenarios as input, and even less are those which support genericity towards their target design model.

The position I take in this paper is that two linguistic models, namely Fillmore's Case Grammar [Fillmore 68] and Grice's Dialogue Model [Grice 75] can be adapted to the definition of instruments responding to both purposes. Each of these two linguistic models may be the core support of a single instrument, the one for automating the identification of requirement model instances out of scenarios, the other one for enriching scenarios and requirements through dialogue. These instruments are described as chunks of methods. That is they are process flows, to which are associated an intention, usage situations, expected targets, and a process to fulfil the declared intention. Both chunks may be reused when textual data like scenarios, or the dialogue in NL are judged to be more convenient communication supports for the RE project. They do not define approaches for validating or eliciting scenarios, but generic processes (generic with respect to the target design model) for the assistance and automation of scenario analysis with linguistic techniques. The purpose of the first chunk is to automate the translation of textual scenarios, provided by the stakeholders, into instances of the design models. This chunk relies on Fillmore's process of semantic analysis of scenarios. Then, a set of heuristic rules defining semantic equivalencies between linguistic and design model patterns must be applied for the identification of candidate instances of the target design model. The purpose of the second chunk is to enrich design models (obtained for example with the first chunk) or scenarios, by handling a dialogue with the stakeholders. To achieve this goal, this chunk can apply Grice's dialogue model and reuse the first chunk to identify new candidate instances of the target design model.

2. Background in Linguistic based Requirement Engineering

Numerous means for the elicitation of requirements with linguistic techniques have been proposed. One can distinguish three kinds of instruments corresponding to the three classical levels of linguistics, namely the lexical, syntactic and semantic levels.

[Goldin and Berry 94] described a method for identifying frequent words or clauses by a statistical analysis technique traditional in the domain of signal processing. If this lexical level method has the advantage of being generic and easy to implement, the lexical knowledge of NL specifications is inadequate for differentiating between different elements of the target design model.

Amongst the other works, many have chosen the syntactic approach to NL analysis. Abott [Abott 83] proposed a technique for developing programs from informal but precise English descriptions. [Chen 83], [Vadera and Meziane 94] [Sykes 95] discussed the links between typical structures of English grammar and ER, VDM and NIAM diagrams, respectively. Another trend was more devoted to the derivation of object oriented design models from specifications in NL [Saeki et al 89], [Kristen 94], [Liang and Palmer 94]. These instruments are also based on correspondences between grammatical functions of the elements participating in a sentence, and the various object models. The variety of target models dealt with suggest that genericity is possible at this level of interpretation of the NL. However, the use of grammatical analysis for identifying several elements having different semantics leads to erroneous results. Indeed, linguists have shown that the semantic role of an element in a sentence must be differentiated from its grammatical function. For example an element having the role of an object moved from a source to a target location may as well have the grammatical function subject as direct object. It is the case of "the form" in Figure 1.

"The central control delivers the form to the ambulance crew"

"The form passes from the central control to the ambulance crew".

Figure 1

This issue has lead the linguists to differentiate between the surface level which is related to the syntax, and the semantic level (or deep level) in which the role of the elements of a sentence may differ from their grammatical function. Fillmore's Case Grammar [Fillmore 68] has been the most influential in categorising semantic roles. After [Bouzeghoub et al 85] [Bouzeghoub 92] who described a process for designing relational databases through semantic models, [Rolland and Proix 92] and [Belkouche and Kozma 93] defined heuristics rules for instantiating design models from typical structures of the case grammar. A related work is [Ohnishi 94,] [Ohnishi 96] who propose to build software requirements in VDRL from textual requirements in Japanese, based on a typology of concepts very similar to the semantic roles of the case grammar.

None of the instruments quoted above proposes a solution adapted to more than one specific design model. Nevertheless the Case Grammar is suitable to the semantic characterisation of any design models as well as to the semantic characterisation of any NL sentence. Genericity might thus be supported by an instrument built on the Case Grammar. Moreover, none of these instruments does address the problem of abstraction, and rare are those which support dialogue for completing and validating the requirement already extracted. Indeed the NL texts inputs of these instruments are supposed to be specifications, that is already abstract, valid and complete requirements.

On the contrary, scenarios are supposed to be partial views and may contain elements belonging to any level of abstraction. When he is faced to such situations, a stakeholder may follow guided processes as offered within intention based computer aided RE tools. However classical techniques of guidance use menu interactions ([Grosz et al 97] , [Si Said and Ben Achour 95]) to capture the stakeholder's intention, which is felt as a constraint to the stakeholder's creativity. A relevant solution seems to be to engage a dialogue in NL with the stakeholder, as suggested by Robertson [Robertson 95] who describes how to handle a scenario-centred dialogue to complete requirements. Using NL dialogue as a strategy to interact with a stakeholder during guidance raises important issues : how should the dialogue be initiated, how to interpret the stakeholder's interventions, how should new questions emerge from this interpretation ?

3. Position to a linguistic support of dialogue and scenario interpretation

As defined in the introduction, my position is that two chunks of method, designed as generic linguistic instruments can provide the adequate frame to support the integration of scenarios into various RE methods. The chunk approach has the advantage of allowing a modularity adapted to the needs of modular RE projects. Moreover, chunks of methods are adaptable; that is to say they are designed to run with several different requirements models. The first chunk I describe underneath handles the activities of completing and validating scenarios through an interactive dialogue with the stakeholders. The model of this dialogue is inspired by Grice's dialogue model, completed with Roulet's functional model of the dialogue. The second chunk performs the interpretation of scenarios and instantiates patterns of any target design model. To handle this activity, Fillmore's Case Grammar is adapted for defining the semantics of NL scenarios and of various design models.

3.1. A chunk for supporting the dialogue with the stakeholder

Linguistic Philosophers like Austin [Austin 62] or Searle [Searle 75] developed the idea that to support the dialogue, functions of the intervention (like assert, authorise, criticise, declare, interrogate, suggest, warn, etc.) had to be recognised on the basis of contextual informations. More generally, the functional model described by Roulet [Roulet 85] gives an account of the links between interventions. Exchange links define complete dialogue interactions whereas complex links gather interventions within which the initial intervention (usually a question) is not answered to by the remaining of the

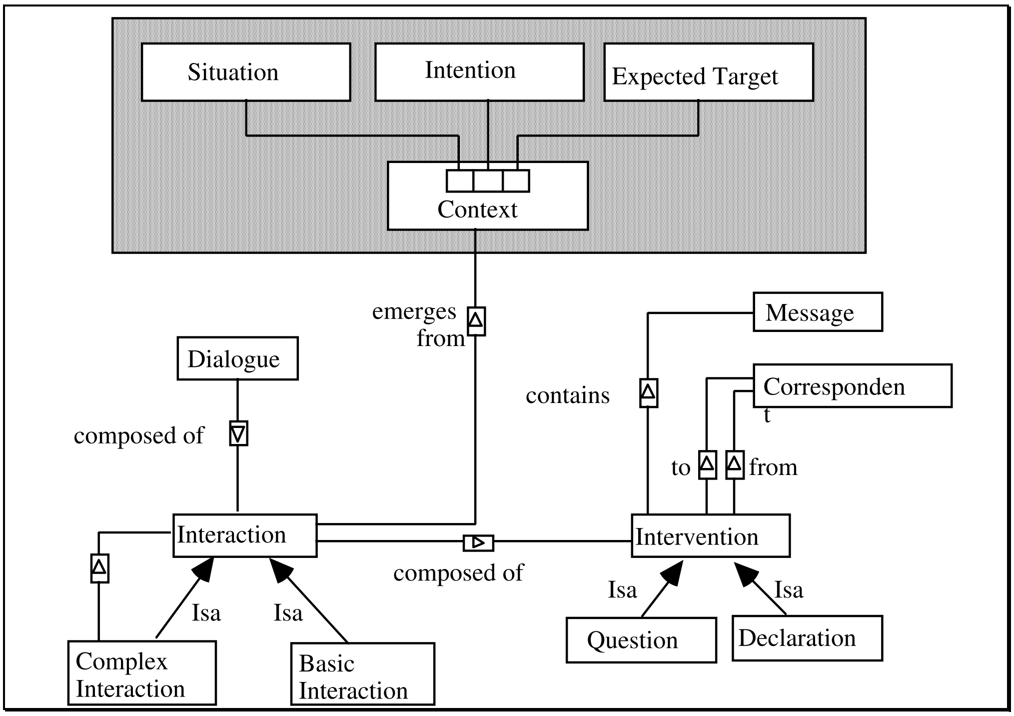

Figure 2 : Dialogue Model

dialogue. Functional modelling defines thus when dialogue should be continued, but does not answer to the issue of choosing how to continue the dialogue.

A solution is at least partially provided by Grice [Grice 75] who underlines the necessity to recognise the speaker's intention. Grice defines a co-operation principle : "any verbal contribution must be such that imposes the general goal of the conversation, at the moment when this intervention is realised". To fulfil this principle, the speaker should follow four groups of maxims (quantity, quality, manner, and relationship) which when violated give rise to new questions about the rational for this violation. Unfortunately, the mechanisms for inferring the new questions are not implementable as such.

Guiding the dialogue with the stakeholder is in some respects similar to guiding other RE processes. The experience within the field of RE has proved that the context of a stakeholder's activity, and especially his intention could be modelled to guide the requirement engineering processes [Rolland 94]. Such model may be adapted to define the process part of the dialogue chunk. The dialogue model I propose in Figure 2 integrates Grice's dialogue model and the functional model to the model of intention based processes. The dialogue is composed of interactions which are complex sequences of *interventions* emerging from the *context* of the dialogue.

The situation which is a part of the context already described in the initial process model is refined into three classes of *situations* . The linguistic situations depend on the linguistic knowledge of the tool. The content situations only define the lexicon of the elements currently part (or proposed as part) of the product. Pragmatic situations include the specification model and identify the product state in the term of this model.

Each intervention is a *message* passing between *correspondents* (the stakeholder or the tool). We define two possible types of interventions : *questions* and *declarations* . Questions are defined as the combination of the *situation* to the dialogue intention. A question-declaration pair makes a *basic interaction* if the declaration is the answer corresponding to the question. However it may happen that the correspondent replies by an other question or by a declaration which does correspond to the expected answer defined as the target in the context. For example, if the RE intention is to identify the state transitions of objects, the question arises "In which state transition is the object O involved ?". To this question the stakeholder might answer "What is a state transition ?", or "I would rather like address state transitions later". In this case of complex interaction, new dialogue intentions may be deduced from the difference between the real answer and the expected answer. Then, new questions arise and the dialogue continues until the relevant state transition has been provided. When the description of the state transition is given, the dialogue is finished, and the translation chunk may be called to propose the equivalent state transition in the terms of the target model.

3.2. A chunk for interpreting scenarios

As defined earlier, the purpose of the translation chunk is to identify candidate instances of the target design model. To realise this translation, the chunk has to perform a semantic analysis of the input NL sentences. As for other instruments quoted above, Fillmore's case grammar was selected for the semantic analysis. This technique has indeed the advantage of providing a structured view on the semantic, and to be adaptable, as shown by the numerous subsequent versions which were proposed ([Shanck 73], [Wilks 77], [Simmons 73], [Boguraev and Spark-Jones 87], [Dik 89]).

The translation chunk, in opposition to the dialogue chunk, is defined by a linear process able to provide results without further intervention of the stakeholder after the initial input of textual scenarios. If the sentence is only a simple clause, the semantic is performed in two steps. First, the syntax of the input sentences is unified with the verb syntax. The sentence is then rewritten in a normalised structure. Second, the normalised sentence may be unified with the semantic structure of the verb, which leads to the emergence of an instantiated semantic pattern for the sentence. If the sentence is composed of several clauses, each clause is first analysed separately as stated above. Then the global structure of the sentence is normalised, and unified with the structure of typical sentence semantic patterns. Once semantic analysis is performed, a set of heuristic rules defining the mapping between case semantic patterns and patterns of the target model are triggered until no new matching is possible.

4. Conclusion

Emerging requirement engineering methods converge towards the inclusion of end stakeholders within the RE process. These new stakeholders are not considered as specialist in RE, are not used to the design formalisms, but are able to express their view on designed system in natural language. It is in this context that scenario based approches and linguistic based instruments were proposed for improving requirement engineering tools and techniques. However, at the same time, RE practitioners ask for more modularity within method, intending at the end to define purposeful RE methods themselves.

This work intends to define a starting framework for responding to both requirements. One the one hand, linguistic instruments are proposed to support dialogue with the stakeholder and automate the analysis of scenarios described in NL. On the other hand, these instruments are presented as generic chunks adaptable to any method, or any scenario based approach to requirement engineering. However previous experiences have proved that both instruments are viable, some issues have to be solved before they were implementable into real tools. In particular the dialogue intentions are not clearly defined at the moment. Likewise, the possible answers of the stakeholders, and the way to identify the new stakeholder's intentions during an automated dialogue have not been tackled. For the translation chunk, the case grammar has now already been largely experimented as a way to identify the semantics of texts like scenarios. More immediate results may be expected in this field, which necessitates to describe the equivalence rules between case patterns and several target model patterns.

Bibliography

[Abott 83] R.J. Abott. Program Design by Informal English Descriptions. Communications of the ACM, 26 (11), pp 882-894, 1983.

[Austin 62] J.L. Austin. How to Do Things With Words. Oxford, 1970.

[Belkhouche and kozma 93] B. Belkhouche, J. Kozma. Semantic Case Analysis of Informal Requirements. Proceedings of the 4th Workshop on the Next Generation of CASE Tools, The Netherlands, pp 163-182, 1993.

[BenAchour and Ralyte 97] C. Ben Achour, J. Ralyte. Experimenting a Framework for Scenario Classification. Under submission at INFORSID'97, 1997.

[Boguraev and Spark-Jones 87] B. Boguraev, K. Spark-Jones. A Note on a Study of Cases. In Computational Linguistics, Vol 13, n° 1-2, pp 65-68, 1987.

[Bouzeghoub et al 85] M. Bouzeghoub, G. Gardarin, E. Métais. Database Design Tools an Expert System Approach. Proceedings of the 11th Conference on Very Large Databases, Stockholm, pp82-95, 1985.

[Bouzeghoub 92] M. Bouzeghoub. Using Expert Systems in Schema Design. In Conceptual Modeling, Database and CASE: an Integrated View of Information Systems, (ed P. Loucopoulos and R. Zicari), Wiley, 1992.

[Carroll 95] J.M. Carroll. The Scenario Perspective on System Development. In Scenario Based Design : Envisioning Work and Technology in System Development. John Wiley and Sons, Ed J.M. Carroll, 1995.

[Chen 83] P.S. Chen. English Sentence Structure and Entity Relationship Diagrams. Information Sciences, 29, pp 127-149, 1983.

[Dick 89] S.C. Dik. The Theory of Functional Grammar, Part I : the Structure of the Clause. Functional Grammar Series, Vol 9, Foris Publications, 1989.

[Fillmore 68] C. Fillmore. The Case For Case. In Universals in Linguistic Theory, Bach & Harms, Chicago, Eds Holt, Rinehart and Winston, pp 1-90, 1968.

[Goldin and Berry 94] L. Goldin, D.M Berry. AbstFinder, a Prototype Abstraction Finder for Natural Language Text for Use in Requirements Elicitation: Design, Methodology and Evaluation. Proceedings of ICRE'94, first International Conference on Requirement Engineering, Colorado, pp 84-93, 1994.

[Grice 75] H.P. Grice. Logic and Conversation. In Syntax and Semantics 3 : Speech Acts, eds Coles and Morgan, Academic Press, New York, pp 41-58, 1975.

[Grosz et al 97] G. Grosz, C. Rolland, S. Schwer, C. Souveyet, V. Plihon, S. Si-Said, C. Ben Achour, C. Gnaho. Modelling and Engineering the Requirements Engineering Process : an Overview of the Nature Approach. Under submission at the RE Journal, 1997.

[Holbrook 90] C.H. Holbrook III. A Scenario Based Methodology for Conducting Requirements Elicitation. ACM SIGSOFT Software Engineering Notes, 15 (1) , pp 95-104, 1990.

[Jacobson et al 92] I. Jacobson, M. Christerson, P. Jonsonn, G. Oevergaard. Object Oriented Software Engineering : a Use Case Driven Approach. Addison Wesley, 1992.

[Jacobson 95] I. Jacobson. The Use Case Construct in Object-Oriented Software Engineering. In Scenario Based Design : Envisioning Work and Technology in System Development. John Wiley and Sons, Ed J.M. Carroll, 1995.

[Kristen 94] G. Kristen. Object Orientation, the Kiss Method: From Information Architecture to Information Systems. pp 97-119, 1994.

[Kyng 95] M. Kyng. Creating Contexts for Design. In Scenario Based Design : Envisioning Work and Technology in System Development. John Wiley and Sons, Ed J.M. Carroll, 1995.

[Liang and Palmer 94] J. Liang, J.D. Palmer. A Pattern Matching and Clustering Based Approach for Supporting REquirements Transformation. Proceedings of ICRE'94, first International Conference on Requirement Engineering, Colorado, pp 180-183, 1994.

[Ohnishi 94] A. Ohnishi. Customizable Software Requirements Languages. Proceedings of the 8th Intenational Computer Software & Applications Conference (COMPSAC'94), Taiwan, 1994.

[Ohnishi 96] A. Ohnishi. Software Requirements Specification Database Based on Requirements Frame Model. Proceedings of 2nd International Conference on Requirement Engineering, Colorado, pp221-228, 1996.

[Potts et al 94] C. Potts, K. Takahashi, A.I. Anton. Inquiry-based Requirements Analysis. IEEE Software, 11 (2), pp 21-32, 1994.

[Robertson 95] S.P. Robertson. Generating Object-Oriented Design Representations via Scenario Queries. In Scenario Based Design : Envisioning Work and Technology in System Development. John Wiley and Sons, Ed J.M. Carroll, 1995.

[Rolland 94] C. Rolland. A Contextual Approach to Modelling the Requirements Engineering Process. In SEKE'94, 6th International Conference on Software Engineering and Knowledge Engineering, Vilnius, Lithuania, 1994.

[Rolland & al 97] C. Rolland, M. Jarke, A. Suttcliffe, E. Dubois, et al. A Framework for Clasifying scenarios. CREWS Deliverable, available on http:://www.rwth-aachen.de, 1997.

[Rolland and Proix 92] C. Rolland, C. Proix. Natural Language Approach to Conceptual Modeling. In Conceptual Modeling, Database and CASE: an Integrated View of Information Systems, (ed P. Loucopoulos and R. Zicari), Wiley, 1992.

[Roulet 85] Roulet. Larticulation du Discours en Français Contemporain. Pete Lang, Berne, 1985.

[Schank 73] R. Schank. Identification of Conceptualizations Underlying Natural Language. In Computer Models of Thought and Language, Shanck & Colby, Freeman, San Francisco, pp 187-247, 1973.

[Saeki et al 89], M. Saeki, H. Horai, H. Enomoto. Software Development Process from Natural Language Specification. Proceedings of the 11th International Confeence on Software Engineering, pp 64-73, 1989.

[Searle 75] J. Searle. Indirect Speech Acts. In Syntax nad Semantics, Coles & Morgan, Vol.3 : Speech Acts, Semina press, New York, pp 59-82, 1975.

[Simmons 73] R. Simmons. Semantic Networks : their Computation and Use for Understanding English Sentences. In Compute Models of Thought and Language, Schanck & Colby, Freeman, San Francisco, pp 63-113.

[Si Said and Ben Achour 95] S. Si Said, C. Ben Achour. A Tool for Guiding the Requirements Engineering Process. In Proceedings of the 6th Workshop on the Next Generation of CASE Tools, Jyvaskyla, Finland, pp 23-42, 1995.

[Sykes 95] J.A.Sykes. English Grammar as a Sentence Model for Conceptual Modelling using NIAM. Information System Concepts, Towards a Considaltion of Views, proceedings of the IFIP international working conference on information system concepts, (ed E.D. Falkenberg, W. Hesse, A. Olivé), pp161-176, 1995.

[Vadera and Meziane 94] S. Vadera, F. Meziane,. From English to Formal Specifications. The Computer Journal, vol 37 no 9, pp753-763, 1994.

[Wilks 77] Y. Wilks. Good and Bad Arguments about Semantic Primitives. Report n° 42, Department of ArtificialIntelligence, University of Edinburgh, 1977.

[Young et al 87]