CREWS Report Series 99 - 07

BRIDGING THE GAP BETWEEN USERS AND REQUIREMENTS ENGINEERING : THE SCENARIO-BASED APPROACH

Camille Ben Achour Mustapha Tawbi, Carine Souveyet

CRI Université de Paris 1 - Sorbonne

90 rue de Tolbiac

75013 Paris - France

{camille, tawbi, souveyet}@univ-paris1.fr

To appear in International Journal of Computer Systems Science & Engineering, Special issue on Object-Oriented Information Systems

Bridging the Gap between Users and Requirements Engineering : the Scenario-Based Approach

C. Ben Achour, C. Souveyet, M. Tawbi,

C.R.I, Université Paris 1 - Sorbonne

90, rue de Tolbiac

75013 Paris - France

{camille, souveyet, tawbi}@univ-paris1.fr

Abstract : *Scenarios have been advocated as a means of improving Requirements Engineering. The assumption is that scenarios allow users to better express their requirements (or more generally their knowledge of the constructed system) because they use their own vocabulary. In Europe, a few scenario-based methods and tools have already been developed in research ; besides, scenarios are widely used in industry. In this context, the long Term Research 21.903 ESPRIT project CREWS has undertaken research on Cooperative Requirements Engineering With Scenarios. This work includes : i. surveys on scenario-based Requirements Engineering approaches and on the use of scenarios in European industry together with ii. the development of four interrelated methods for the acquisition and validation of requirements based on scenarios. This paper reports the CREWS objectives and achievements, and further details the CREWS-L'Ecritoire approach, which purpose is to guide the elicitation of requirements by exploiting textual scenarios. Pointers to the CREWS surveys, to the four interrelated CREWS approaches, and to on-going works undertaken by CREWS are also given.*

1. Introduction

The purpose of *Requirements Engineering* (RE) is to lead to the realisation of the users' and customers' needs for a future system. During the RE process, these needs are elicited, negotiated, validated, specified and documented in requirements documents.

To develop a quality system necessitates quality system requirements, i.e. system requirements that are not incomplete, inconsistent, or that do not correspond to the users and customers' needs. Therefore, RE is an important phase of a system development project. Underestimating this importance may lead to aborted projects, overrun budgets, or system crash, as numerous past experiences have shown [Johnson 95].

Whereas conventional methods were devoted to languages and models to express requirements, the RE community has perceived RE as a process that goes beyond the production of complete and consistent system requirements specifications.

RE is also *« a cooperative and iterative learning process »* [Opdhal 98]. As an *iterative learning process*, RE relates closely to elicitation, validation, tracing, revision, specification and reuse. Moreover, as a *cooperative process*, requirements elicitation involves multiple stakeholders who have to communicate and understand each other : users, customers, domain experts, project managers, designers of previous system releases, etc. Many of these stakeholders have no RE skills. They do not know how to perform, and specification languages and models are not part of their vocabulary. There is therefore a *'gap'* between users (who have needs but are unaware of how to specify them as requirements) and requirements engineers (who know how to specify requirements, but do not know the needs).

In reaction to this issue, the RE community has recently shifted its centre of interest on techniques oriented towards users. The s*cenario-based RE approach* belongs to this family of user-oriented RE techniques. The scenario-based approach, of which CREWS[1] is part of makes the assumption that a mutual understanding between users and Requirements Engineers may be ground in scenarios. In the specific case of CREWS, scenario-approaches aiming at requirements elicitation and validation have been developed. In these approaches, scenarios are used as a communication medium allowing requirements engineers better understand the usage requirements, and the other way round, allowing to users better understand the requirements specifications.

The purpose of this paper is twofold : first, to make an general presentation of the research objectives and on-going works undertaken by CREWS, and second to detail the CREWS-*L'Ecritoire* approach for requirements elicitation based on textual scenarios. These aspects are respectively tackled in section 2 and section 3. Section 4 discusses follow-ups and research issues currently dealt with in the CREWS project.

---

[1] CREWS (Cooperative Requirements Engineering With Scenarios) is a long-term research project funded by the European Community (21.903) of the ESPRIT 4^{rth} framework programme.

2. The CREWS scenario-based approach to RE

Scenarios have been used to support mutual learning and promote creativity, as for example in SystemCreativity by Intel, or in Scenario Engineering by Xerox Parc. However these approaches exist in separate communities of practitioners and researchers and have not found their ways into RE methods.

In this context, the objective of CREWS is to « *develop, evaluate and demonstrate the applicability of innovative methods and tools for cooperative scenario-based requirements elicitation and validation* » [Jarke 96]. Two research directions were taken in CREWS : i. the survey of research and industrial practice of scenario-based RE, and ii. the development of four individual but interrelated scenario-based RE approaches. These are respectively presented in the two next subsections.

2.1 The CREWS surveys on scenario-based RE

CREWS has undertaken two surveys on scenario-based approaches in RE, Human Computer Interface, Software Engineering and Information System Engineering. The first one is a state of the art on scenario-based methods and tools developed by research [Rolland 98], and the second is a survey of the practice of scenarios in European industrial organisms [Weidenhaupt 98]. These are commented in turn.

2.1.1 State of the art of scenario-based methods and tools in research

The CREWS state of the art on the scenario-based approaches covers over 40 different research approaches out of which 12, selected for their representativeness of the main trends, are studied in the details [Rolland 98].

This survey exhibits a great diversity of scenario-related methods, models and notations. Indeed, scenarios support the capture of examples, scenes, narrative descriptions of contexts, cases of system use and illustrations of agent behaviours. Moreover, they allow to elicit requirements in envisioned situations [Potts 94], to help in the discovery of exceptional cases [Rolland 98a], [Sutcliffe 98], to derive conceptual object-oriented models [Dano97], [Rumbaugh 91], [Jacobson 95], [Rubin 92], to understand needs through scenario prototyping [Hsia 94] and animation [Lalioti 95] to reason about design decisions [Carroll 95], [Young 87], create context for design [Kyng 95] and so on.

The underlying reason for the popularity of scenario-based approaches seems to be that people react to descriptions of real happenings and real things. However, one consequence is a large diversity in approaches and definitions, and the difficulty to precisely define what a scenario is. In general terms, a *scenario* can be defined as « *a sequence of actions or events for a specific case of some generic task that a system is meant to accomplish* » [Plihon 98]. It consists of a behaviour and a context describing the agents, objects, and environmental setting of the system under development [Jarke 96]. However, in a more precise way, scenarios were developed [Rolland 98] for different *purposes* with different *contents*, expressed in different *levels of abstraction* and with different *notations*.

In so far as their *purpose* is concerned, scenarios can be *descriptive*, *explanatory* or *exploratory*. *Descriptive* scenarios [Potts 94] capture requirements by enabling the analyst and users to walk through a process and understand its operations, actors, the events triggering the process etc. Thus, descriptive scenarios aid in the clarification of how a process performs, who are the involved parties and how the process is activated as well as the conditions under which it is activated. *Explanatory* scenarios [Wright 92] raise issues and provide rationale for these issues. They identify why something happens in the real world, what leads to it, what are its causes, what are commonly occurring events which require handling etc. Through this the attempt of explanatory scenarios is to describe the desirable features of the system to be developed. Finally, *exploratory* scenarios [Holbrook 90] are useful when different possible solutions for satisfying a given system required exist. These solutions are to be examined and evaluated to arrive at the right solution. Such scenarios establish a direct link between requirements and desired solutions.

As mentioned above, scenarios have different *contents*. This can be behavioural information identifying the actions, activities, events carried out in the usage world; a description of the objects of the real world together with their attributes; events and event histories; organisational information like the structure of the company, the groups, departments and agents found in it, or even stakeholder information including the characteristics of people, their views and aspirations [Nardi 92]. However, scenarios may also concentrate on the required functional features of a system [Firesmith 94], [Glinz 95], [Olle 92], [Potts 94], [Rubin 92], [Rawsthorme 96], [Somé 96].

Scenarios can be expressed at three different *levels of abstraction* : instance, type and mixed. In the former case [Carroll 95], [Potts 94], [Young 87], a scenario uses specific names or events with real argument values. These scenarios describe particular instances of use which can form the basis for discussion of what happens, why and how. Type scenarios [Hsia 94], [Jacobson92] do not use individual entities but entity types. Thus they do not refer to Smith but to customers. Each execution of a type scenario is an instance scenario. Finally, mixed scenarios [Rolland 98a] are those that have some parts at the instance level and others at the type level.

Finally, scenarios can be expressed using different *notations* ranging from the *informal*, *semi-formal* to the *formal*. *Informal* scenarios use natural language [Rolland 98b], [Erickson95] , [Holbrook 90], videos [Wood 94], [Haumer 98] story descriptions, and are valuable in those cases where the user community is unwilling (or unable) to deal with formal notation. *Semi-formal* scenarios use a structured notation like tables [Potts 94] and scenario scripts [Rubin 92] in capturing real activities. Finally *formal* scenarios are expressed in modelling languages based on regular grammars [Glinz 95]or state-charts [Harel 87]. They are useful to run as simulations to present a vision of what the future system will look like and to gauge user reactions to it.

In the CREWS survey of research scenario-based approaches [Rolland 98], four views on scenarios, namely : purpose, contents, lifecycle and form are developed and used as a classification framework allowing to characterise and position different approaches out of which the ones quoted above.

2.1.2 Survey of the practice of scenarios in the European Industry

To complement the more research-oriented classification framework, the CREWS team has undertaken 15 site visits to industrial projects in four European countries [Weidenhaupt 98]. Each site visit was conducted by a structured interview involving two or three people from the CREWS project and one or two members of the project examined. The structured interview was based on a catalogue of questions on the scenario characteristics which were derived from the CREWS classification framework. Thereby the general distinction between the four main views proved to be very useful to structure the interviews and the observations from real world practice. Concerning the four views the main findings of the site visits are the following:

Form view : the three *description media* which dominated in the projects are *text*, *graphics* and *image*. The majority of the projects employed natural language either in prose text or in structured text following a more or less rigid template or table-structure. To a less extent, also unstructured pictorial representations (images, sketches) and semi-formal diagrammatic notations (e.g. messages trace diagrams) were observed. In addition there is a correlation between the form and contents. Natural language is predominant for context and interaction scenarios whereas diagrammatic notations are preferred in scenarios depicting the system internal. The *animated presentation* was also used in practice.

Contents view : in practice, scenarios seemed to be mainly treated as means to clarify interactions between the system and its users. Nearly all the projects use scenarios to capture system-user interactions whereas only one third also considered the system context and/or the system internal interactions, respectively. *Argumentation* was not significantly used in scenarios whereas *mixed abstraction* seemed to be uniformly used.

Purpose view : most surprisingly was the richness of *usage* observed. Besides the purposes already identified by the classification framework, the use of scenarios for concretizing abstract models, for improving agreement and consistency, for reducing complexity, and for complementing prototypes and glossaries was emphasized.

Life cycle view : in all projects examined the fact that scenarios are artifacts which evolve over time and must therefore be managed accordingly was mentioned. The question raised in the *lifespan* facet, namely 'are scenarios *transient* or *persistent* items?' was really a concern in almost all projects. In addition, the site visits raised issues like how to impose partial views on scenarios, how to develop and maintain scenarios in a large-scale distributed setting, how to review scenarios, how to ensure traceability of scenarios to other software artifacts.

2.1.3 ***Lessons learned from the CREWS surveys***

The diversity of scenario form, contents, purpose and life cycle presented in the review of academic approaches was greater in industry than one would expect from the UML-incited understanding of scenarios and use cases. It was, however, interesting to notice that the focus of academia on some particular aspects of scenarios shifted during our investigation. For example, form issues play in practice a much minor role than in research literature while the usage and life-cycle aspects were much richer than anticipated from the literature survey. It is clear from our surveys that there exists some divergence between method recommendations in the literature and the practice of scenario based approaches. The analysis of this divergence raised six key issues surrounding scenario based approaches today:

i. *Methodological guidance :* the lack of both formal product models and guidelines to support the process of developing scenarios was observed in the literature survey and emphasized in practice. Users request more explicit methodological guidance and more adequate tool support.

ii. *Managing scenario evolution :* the issue is on one hand, to provide better means to structure a large collection of scenarios and their relationships with other artifacts and on the other hand, to support their changes over time.

iii. *Scenario authoring :* as natural language is the most widely used means for expressing scenarios there is a need to support the writing of textual scenarios in order to overcome the risks raised by the use of NL. Style guidelines, templates, natural language analysis support etc. are possible contributions to solve this issue. The definition of such writing rules was a high preoccupation of practitioners in the examined projects.

iv. *Clarifying the role of scenarios :* there is an explicitly expressed need for clarifying, *when*, *why*, *how* which type of scenario can be used to support which type of design activity. Solutions to this issue will be a support for a mix and match approach to producing a customized scenario based approach for a particular project.

v. *Impact of scenarios on Non Functional RE :* taking into account non functional requirements in RE is an important issue raised among others by the CREWS industrial steering committee. It seems that scenarios can contribute to this issue by providing a concrete way to visualize the impact on non functional requirements onto system and organizational context requirements. How they can contribute to non functional RE is a research question.

vi. *Usefulness evidence :* although practitioners expressed their interest for continuing using scenarios in the future scenarios were also judged dangerous and costly in the context of large scale projects. Collecting evidences of the usefulness of scenario based approach is a real issue.

Based on these lessons learned, CREWS has undertaken the development of RE scenario-based approaches. These are described in the next subsection.

2.2 The four CREWS approaches

Four related methods and tools for cooperative scenario-based requirements acquisition and validation were developed in CREWS. One of the strengths of the four CREWS approaches is that they have been designed to complement each other. As a result, CREWS can offer a coherent scenario-based RE method with different techniques and software tools to achieve each process. Furthermore, this method also includes existing approaches such as use case analysis [Jacobson 92] and notations such as the UML [Rumbaugh 96]. Two of the CREWS approaches guide requirements acquisition.

i. Existing approaches do not give method guidance to represent rich descriptions, or scenes, of current system use and transform these descriptions into conceptual models. The first approach guides co-operative elicitation of system requirements from scenes recorded in multimedia representations such as video footage and audio recordings, to elicit different types of model and system requirement from these. A method and multi-media software prototype support this process representations [Haumer 98].

ii. The second approach guides the semi-automatic extraction of system requirements from natural language descriptions of scenarios and use cases. The process encourages an author using CREWS guidelines when writing a use case/scenario to ensure both its completeness and correctness, as well as making the use case/scenario amenable to computational analysis. A software tool then applies a set of rules to the use case/scenario description to extract candidate system requirements [Rolland 98a].

The other two approaches guide requirements validation.

iii. Scenarios provide useful 'test scripts' for a requirements specification, however there is a lack of process guidance for systematic scenario generation and use. The third approach provides a method and software tool to generate useful scenarios then walk users through these scenarios, as well as to semi-automate the detection of missing or incorrect system requirements through computational analysis of these scenarios [Sutcliffe 98].

iv. The fourth process validates system requirements through the animation of scenarios derived from a formal specification of the system. This language is compatible with the

ALBERT agent-based requirements modelling language, thus enabling easy inter-linking of requirements specifications, declarative non-deterministic scenario scripts and deterministic scenario execution traces as a basis for requirements animation and hence validation. This approach is also supported by a software tool [Dubois 99].

The next subsection details the CREWS-*L'Ecritoire* approach, above mentioned at (ii.).

3. Overview of CREWS-*L'Ecritoire*, an approach to requirements elicitation based on textual scenarios

In the following, we get a brief insight into the essence of the CREWS-*L'Ecritoire* approach before entering into a detailed description of its objectives and features.

3.1 The essence of CREWS-L'Ecritoire : coupling goals and scenarios

Whereas initial research efforts focused on the definition of system requirements[Spivey 92], [Jones 90], [Gutttag 93], [Rumbaugh 91], [Jacobson 95], recent attempts have been made to develop approaches that support the elicitation of requirements better meeting the organisation's goals and needs.

The argument of goal-driven approaches is that the rationale for developing a system is to be found outside the system itself, in the enterprise in which the system shall function [Loucopoulos 94]. RE is therefore concerned with the elicitation of high-level goals to be achieved by the envisioned system [Ant96, Bub94, Dar93], the refinement of these goals [Dardenne 93], [Yu 94], [Rollland 98a] and their operationalisation into system requirements specifying how goals should be accomplished by the proposed system [Antòn 96]. However, practical experience shows that :

(a) goals are not given and thus, goal discovery is not an easy task [Rolland 98], [Antòn 96],

(b) application of goal reduction methods [Dardenne 93] to discover component goals of a goal is not as straight-forward as the literature suggests [Bubenko 94], [Elektra 97], [Antòn 96] and

(c) eliminating uninteresting and spurious goals is necessary and difficult [Potts 97].

Independently of goal modelling, the scenario-based approach to RE arguments that typical scenarios are easier to get in the first place than goals. Goals can be made explicit only after deeper understanding of the system has been gained. The industrial practice survey conducted by the CREWS consortium confirms that scenarios are useful in particular when abstract modelling fails [Weidenhaupt 98].

In order to overcome some of the deficiencies and limitations of goal-driven and scenario-based approaches used in isolation, some proposals have been made recently to couple goals and scenarios together. In [Dano 97], [Jacobson 95], [Leite 97], [Pohl 97] goals are considered as contextual properties of scenarios, and in [Cockburn 96] they are used as a means to structure them. The goal-scenario combination has been used to operationalise goals [Antòn 96], [Holbrook 90], [Potts 94], [Rolland 98a], to check whether or not the current system usage captured through multimedia scenarios fulfils its expected goals [Haumer 98], to infer goals specifications from operational scenarios [VanLamsweerde 95] and, in the CREWS-*L'Ecritoire* approach, to *discover new goals through scenario analysis* [Rolland 98a].

The CREWS-L'Ecritoire approach [Rolland 98a], [Rolland 98b] uses a *bi-directional goal-scenario coupling allowing movement from goals to scenarios and vice-versa*. The global solution is in two parts : when a goal is discovered, a scenario can be authored for it and once a scenario has been authored, it is analysed to yield goals. By exploiting the goal-scenario relationship in the reverse direction, i.e. from scenario to goals, the approach pro-actively guides the requirements elicitation process.

The following subsection presents in-depth the issues CREWS-*L'Ecritoire* aims at solving by exploiting this bi-directional goal-scenario coupling.

*3.2 Objectives of the CREWS-***L'Ecritoire** *approach*

Before entering into the details of the CREWS-*L'Ecritoire* objectives, it is necessary to notice that the approach makes the choice of using scenarios expressed in the *textual* form.

Though rather recent, the use of textual scenarios, in requirements elicitation already occupies an important place in industrial practice but raises several issues [Weidenhaupt 98] : practitioners need more methodological advice on how to write textual scenarios, to organise them, to move towards formal representations, to elicit requirements out of them. They also need ways to identify textual scenarios, to check their correctness, their consistency, completeness, etc. Based on the survey of industrial practice undertaken by CREWS [Jarke 97], we classify the issues underlying requirements elicitation approaches based on textual scenarios in five generic groups. CREWS-*L'Ecritoire* aims at tackling the difficulties related to each of these five groups of issues.

i. *The difficulty with scenario authoring* : the CREWS study of industrial practice [Weidenhaupt 98] confirms that in most cases scenarios are actually textual. *Authoring scenarios* consists of not only writing textual scenarios, but also of transforming them, for example to correct the numerous errors they can contain because of their informal nature. Indeed, scenarios being informal texts, what differentiates them from other kinds of text ? How does one verify that textual scenarios are correctly written, that they do not contain ambiguities, inconsistencies ? How even to verify that it is actually scenarios that are written ? These are the kinds of problems underlying the scenario authoring issue.

ii. *Going from informal to formal scenarios* : the informal nature of textual scenarios, if convenient to human expression, is problematic in the sense that quality RE can only be controlled through formal means. The questions underlying the *informal to formal scenario* definition issue are the following : how to ensure that it is the expected kind of artefact that is written ? What are the relationships between the informal expression of a scenario and its formal definition ? Solving these problems is necessary to support, for instance, computer based automated reasoning on scenarios.

iii. *The black box - white box issue* : in practice, it was proved that stakeholders like to apply the *black box - white box* principle. The difficulty with this principle is to ensure that the level of abstraction of the black box is preserved when its inside is being described. For example, [Jarke 97] shows that in many projects, different levels of detail and different concerns are mixed up in the same scenario description. The issue covers the definition of levels, as well as ensuring that these levels are respected when authoring scenarios.

iv. *The methodological guidance issue* : practitioners miss *methodological guidance*. Whereas the literature provides extensive descriptions of scenarios as products, these approaches do not tell us what to do with scenarios, when to do it, and how to do it. It is largely admitted that such guidance should be provided with a process model. However, most often guidance is based on informal and manual heuristics rather than on enactable guidelines.

v. *The scenario fragmentation issue* : organising scenarios, or identifying new scenarios to be authored within a collection of existing scenarios are the problems raised by the *scenario fragmentation* issue [Cockburn 96]. The black box-white box principle allows the organisation of scenarios into levels and therefore provides a partial solution to the fragmentation issue. However, this is clearly insufficient as scenarios at a level themselves need to be organised.

vi. *The scenario - requirement relationship* : scenarios are only fragments of knowledge of the system for which the requirements are elicited. The relationships between several scenarios, or the relationships between scenarios and system requirements are very seldom explicit in the scenario definitions. The *scenario - requirement relationship* is thus not obvious : are scenarios requirements ? If not, how do they relate to them ? Are they issued from requirements or the other way round are requirements part of the scenarios ?

The combination of this rather complex set of objectives drove the development of the CREWS-*L'Ecritoire* process model. The two major objectives emphasised in this process model are :

- delivering an organised and complete collection of goals that specify the system requirements and that are operationalised by scenarios written in natural language, and
- supporting its production by a stepwise and guided process which progressively transforms initial and partial scenario descriptions in natural language prose into well structured and non ambiguous texts integrated into the goal-scenario collection.

The next section describes how these objectives can be met according to the CREWS-*L'Ecritoire* process model.

3.3 Overview of the CREWS-L'Ecritoire approach

The CREWS-*L'Ecritoire* requirements elicitation process consists in the top-down production of collections of goal-scenario pairs organised hierarchically. This incremental process is informally depicted in Figure 1. As the figure shows, the approach aims at *helping* users and requirements engineers to :

- describe in *textual scenarios* possible uses of the system; this shall referred to as *scenario authoring*,
- integrate scenarios and system requirements into *scenario collections*, this shall be referred to as *scenario organisation*, and

– discover *requirements/goals* by analysing the scenarios of a goal-scenario collection, this shall be referred to as *scenario analysis*.

Therefore, in CREWS-*L'Ecritoire*, the discovery/elicitation of requirements is effectively achieved through a bi-directional coupling of goals and scenarios allowing movement from goals to scenarios and vice-versa. As already defined, the goal-scenario coupling is exploited in the forward direction from goals to scenarios by authoring a scenario each time a goal is discovered, and in the reverse direction by yielding goals from the analysis of each authored scenario.

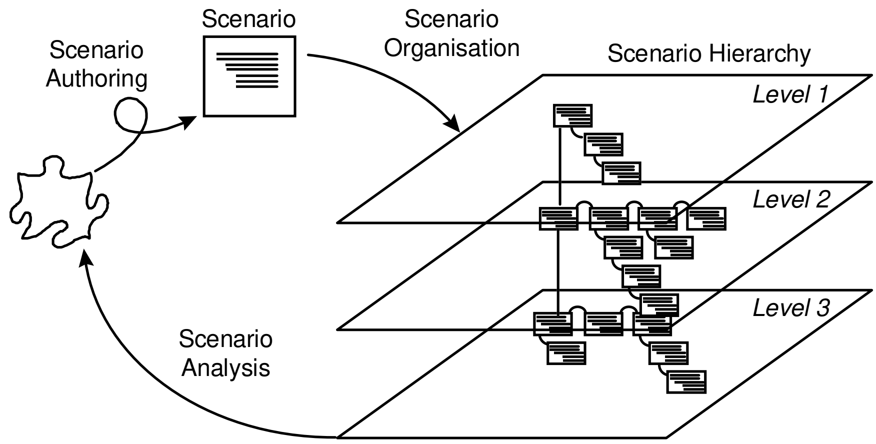

Figure 1 : overview of the scenario-based requirements elicitation process

The choice of the textual form for scenarios makes *scenario authoring* critical. The CREWS-*L'Ecritoire* approach looks for a middle ground between completely free mode of use of natural language which increases the risks of errors, and predefined templates which, we believe, constrains too much the behaviour of the scenario author. It combines the use of *informal narrative prose* to express scenarios with *structured natural language* which is necessary for scenario analysis. The scenario authoring step is thus itself a complex process during which an informal short narrative is *written* then *conceptualised* into a complete, non-ambiguous and structured formal description [BenAchour 99].

Scenarios being *'fragmentary pieces of system knowledge'*, they always belong to a set. CREWS-*L'Ecritoire* proposes to address the scenario fragmentation issue by automatic *organisation* of hierarchical goal-scenario collections [BenAchour 98]. Such hierarchies are defined at different levels. A scenario, presented as a black box at a given level can be seen as a white box at a lower level. The scenario organisation is thus compatible with the black box-white box principle.

Once a scenario is authored and integrated into a scenario hierarchy, CREWS-*L'Ecritoire* guides its *analysis* to help discovering new requirements [Rolland 98a]. In the approach, the requirements are represented as *goals*. Each scenario is authored for a given goal. Therefore, the set of goals that are coupled to the scenarios of a scenario hierarchy define the system requirements. Requirements are thus systematically elicited by completing goal-scenario hierarchies with new goal-scenario pairs to be organised, and in which the scenarios need to be authored and in their turn analysed.

3.4 The CREWS-L'Ecritoire process model

CREWS-*L'Ecritoire* guidance is based on methodological facilities which systematise the process. Each facility provides a different strategy to perform a step in the process. Defining the CREWS-*L'Ecritoire* requirements elicitation process necessitates thus to tell which steps can be performed, when they can be performed, and what are the different strategies available to perform them. These information are formalised in the *process model* shown in Figure 2.

This process model is presented as a map, i.e. a directed graph with *intentions* (the ones that the stakeholder has in mind) as nodes and *strategies* as edges [Rolland 99]. The directed nature of the graph shows which intention can follow a given one. Strategies advise on the different manners to achieve a selected intention once another one is achieved. The set of strategies relating two given intentions is called a *section*. There are in addition two distinct intentions, respectively called *start* and *stop*, that represent the intentions of starting navigating in the map and stopping doing so. The scenario organisation feature is not part of the graph; indeed, it is included in the goal elicitation strategies and can be systematically triggered in an automated way.

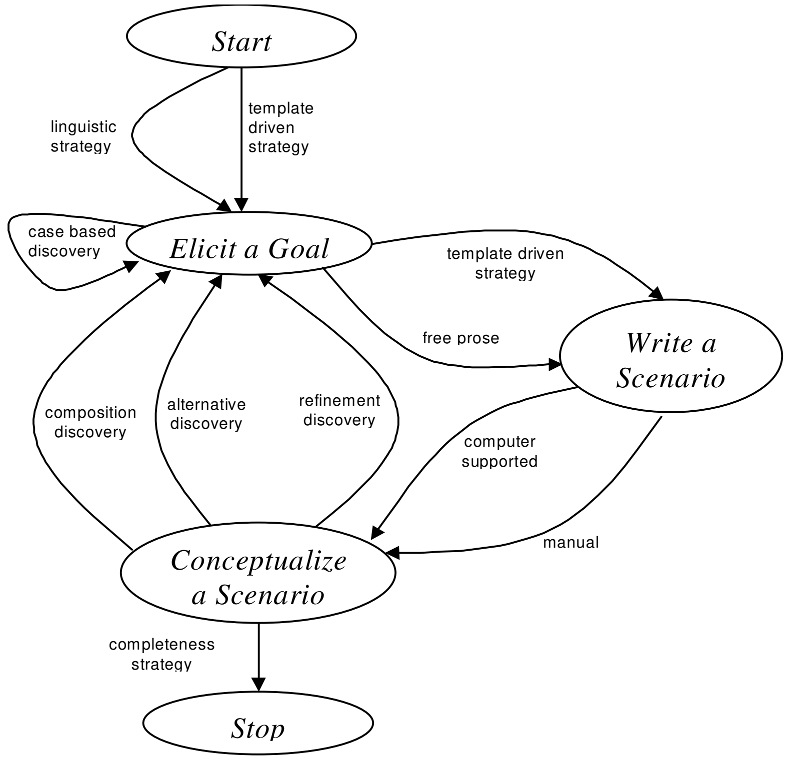

Figure 2 : formal definition of the CREWS-*L'Ecritoire* process model

The following subsections detail the four major sections of this process model.

3.4.1 Initial goal elicitation

Before entering into the goal elicitation-scenario authoring loop, it is necessary to identify the top level goal of the hierarchy to be constructed. The *Linguistic, template-driven* and *case based discovery* strategies for goal elicitation make use of a linguistic definition of goals [Rolland 98a]. According to this definitions, any goal can be expressed with one verb and several parameters, each parameter playing a different role with respect to the verb. For example, the goal :

'(Provide)$_{Verb}$ (recycling facilities)$_{Object}$ (to our supermarket customers)$_{Destination}$'

Can be considered as a high level goal assigned by an enterprise to a new system. Linguistically, this goal includes a verb ('to provide'), and two parameters : an object (*the service which* is provided by the system) and a destination (*the person to which* the service is provided by the system).

The template provided in the *template driven* goal elicitation strategy is based on the aforementioned linguistic goal definition.

The *linguistic strategy* exploits the semantic relationship between verbs and goal parameters. The idea is to ask for the goal verb and to propose the list of possible parameters to be completed.

Once an initial goal has been identified, it is possible to search for design alternatives. To help identifying possible design alternatives, a *case based discovery strategy* is thus proposed. The rule guiding this goal elicitation strategy is based on goal analysis; its implementation in the CREWS-*L'Ecritoire* prototype is shown in Figure 3. First, it is necessary to identify the structure of the analysed goal (as shows the right part of the frame named 'Goal Structure'). Then, for each of the parameters identified in the analysed goal, a set of alternatives has to be proposed (right part of the same frame). Third, the rule combines all the proposed alternative parameters thus generating a collection of new goals. For example, starting from the goal :

'Provide recycling facilities to our supermarket customers'

the rule helped generating the alternatives :

> *'Provide bottle recycling facilities to all customers with a card based machine',*
> *'Provide bottle recycling facilities to our customers with a card based machine',*
> *'Provide paper recycling facilities to all customers with money return machine',* etc.

Goals of interests can then be elicited (this is the case of the two last goals in the list above), and if necessary rephrased.

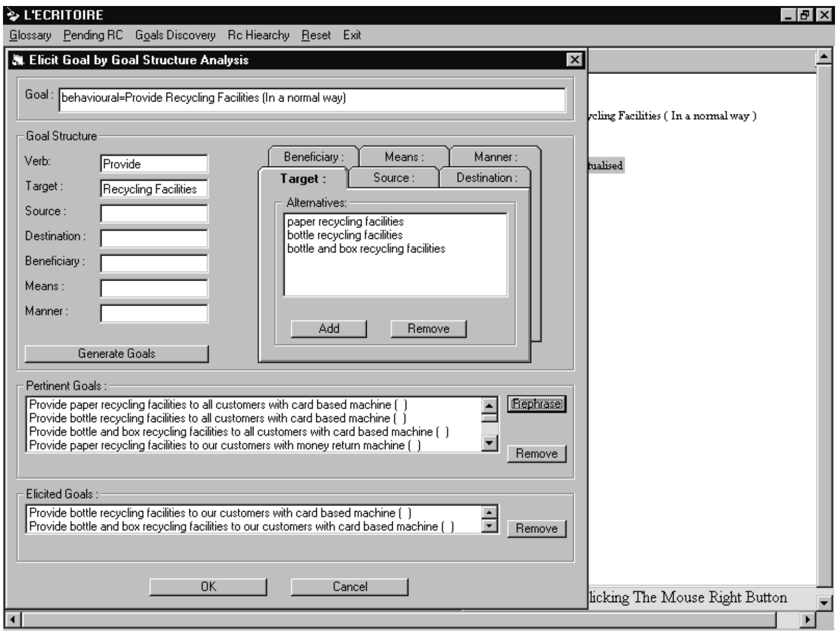

Figure 3 : example of goal discovery using the cased based strategy

Once a goal has been elicited, a scenario can be authored for it. The authored scenarios should conform to the formal definition provided by the *conceptual scenario model*. This model defines the coupling relationship between scenarios and goals, the levels at which scenarios can be described, the set of concepts used in scenario descriptions and the relationships between these concepts [Rolland 98a], [BenAchour 99].

Before being conceptualised according to the conceptual scenario model, scenarios have to be written. Thus, as the process model presented in Figure 2 shows, scenario authoring is guided in two steps : first, scenarios are written, then they are conceptualised.

3.4.2 Scenario writing

CREWS-*L'Ecritoire* proposes two strategies for writing scenarios : a *template driven strategy* and a *free prose strategy*. Both strategies are based on the *linguistic scenario model* which defines from the linguistic perspective the restricted subset of natural language that can be used in texts describing scenarios. This model borrows a Chomskian perspective and tackles the surface level (at which the syntax of the language used in scenarios is defined) and the deep level (which deals with the semantics of the scenario clause and sentences). At both levels, the linguistic scenario model is tightly related to the precise semantics defined for scenarios in the conceptual scenario model [BenAchour 98a], [BenAchour 99].

Free prose scenario writing is guided by *informal guidelines* which help : limiting the size of the input scenarios, targeting their content, advising on what is expected at this very moment in the process (according to the linguistic scenario model), and providing, if necessary, templates to write sentences. The underlying assumption is that scenarios can be more easily transformed and lead to more correct analysis if they are written according to these informal guidelines. Figure 3 presents an example of scenario written with the CREWS-*L'Ecritoire* prototype tool using the free prose strategy.

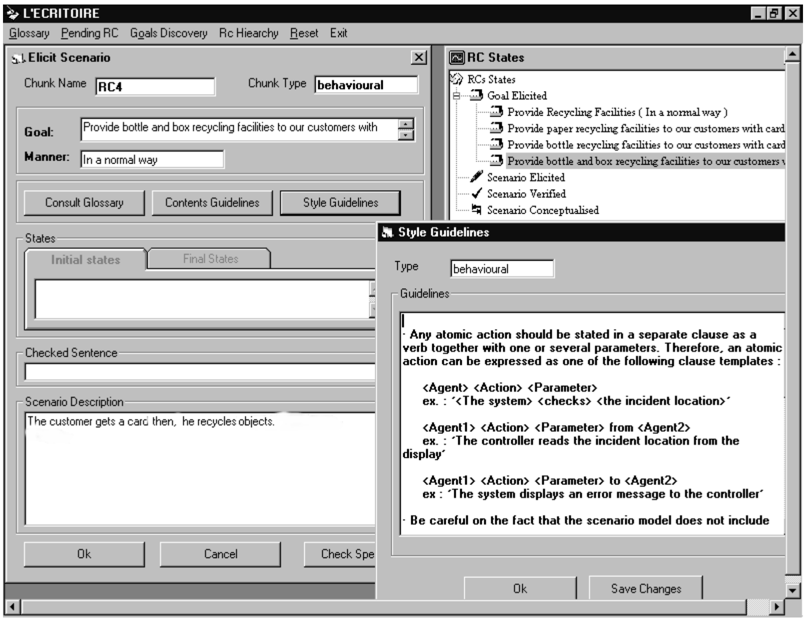

Figure 4 : example of scenario writing using the CREWS-*L'Ecritoire* tool

3.4.3 Scenario conceptualisation

The transformation of scenarios is based on *semi-automated guiding rules*. These guiding rules work as systematic methodological aids. Contrary to the informal guidelines, they are defined formally and automated in the tool, though interaction with user is needed.

The rules guiding the *conceptualisation* of textual scenarios support their linguistic analysis, correction and their mapping onto formally structured concepts. This permits the approach to take advantage of free text expression while limiting the complexity of their interpretation [BenAchour 98a], [Rolland 98b]. In addition to the computer supported strategy, a *manual strategy* is also proposed for conceptualising scenarios. This strategy looses the advantage of automation and requires an in-depth knowledge of the scenario models. However, we believe several levels automation should be provided in the process guidance so as to let the user decide on his own on his best practice. Similarly, an alternative scenario writing strategy is proposed : the *template driven strategy*. This strategy is more constraining than the free prose strategy as is requires to use systematically predefined templates. This strategy can be found less convenient as it requires much more interactions with the tool during scenario writing. However, it has the advantage of recalling transparently the linguistic nature of scenarios, thus minimising the risk of divergence from the scenario model.

Once conceptualised, scenarios can be analysed to yield news goals. For each of these goals, new scenarios will be written and so on.

3.4.4 Goal elicitation, based on scenario analysis

One important part in the CREWS-*L'Ecritoire* process model stands in the provision of different strategies to support *goal elicitation*. Upon the completion of scenario conceptualisation, any strategy can be dynamically chosen. Thus, there is no statically imposed linear order in the flow of activities. This flexibility in strategy selection is the main advantage of the formalism used to model the CREWS-*L'Ecritoire* process.

Each of the goal elicitation strategy results in a different kind of structure between the discovered goals and the goal associated to the analysed scenario. Three kinds of structures are used to relate goals in a hierarchy (the *AND*, *OR*, and *Refinement* relationships); three strategies are thus proposed.

The *composition strategy* helps finding goals that are complementary to the one to which the analysed scenario is attached. It is thus associated to the *AND* relationship between goals. The goals discovered using this strategy tend thus towards a complete definition of the system.

The *alternative strategy* helps finding possible ways to achieve the same goal as the one to which the analysed scenario corresponds. It is thus associated to the OR relationship. The scenarios associated to the collection of goals discovered using the alternative strategy tend thus towards a complete description of the possible uses of a given system function.

The *refinement strategy* helps finding goals at a lower level of abstraction than the one attached to the analysed scenario. This strategy exploits the refinement relationship according to the black-box / white box principle.

Each goal elicitation strategy is supported by formally defined rules. Figure 4 shows for example the use of the CREWS-L'Ecritoire implementation of a refinement rule. The rule proposes to consider every atomic action in the analysed scenario as a goal. The list of atomic actions of the analysed scenario is thus provided to the user (see the 'Action list' frame in the 'Elicit refining goals' window). Selecting one action in that list opens the 'Elicit goal' window in which the user is requested to rephrase the selected action as a goal.

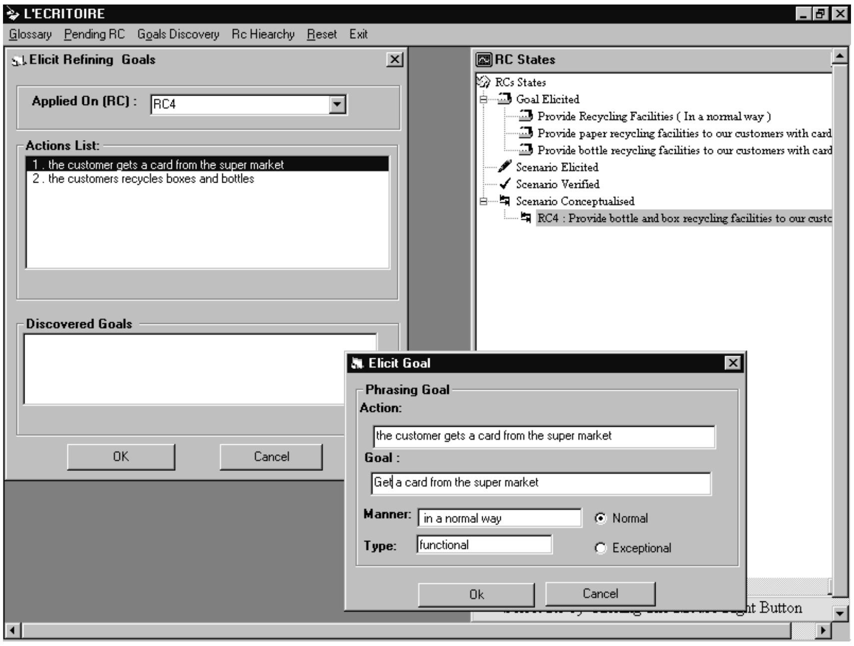

Figure 5 : example of goal elicitation using the CREWS-*L'Ecritoire* tool

The complete set of rules proposed to guide automatically goal elicitation is given in [Rolland 98a], [Rolland 98b].

4. Conclusion

We hope to have demonstrated in this paper that the applicability of scenario-based techniques extends far beyond the standard use of use cases in Object-Oriented techniques. We presented the issues - in academia, as well as in industry- underlying the scenario-based RE techniques. The CREWS project, based on a comprehensive study of literature and practice, has selected the most critical open issues and developed methods to solve them. One of the CREWS approaches, namely the CREWS-*L'Ecritoire* approach, was extensively described in this paper.

Two research directions are currently tackled by the CREWS team [Jarke 99]. On the one hand, the individual scenario-based approaches are being evaluated, so as to demonstrate their effectiveness. The evaluations are based on empirical studies (e.g. [BenAchour 99a] [Tawbi 99]), on application of the CREWS approaches to case studies at the industrial scale (e.g. [Rolland 99a]), and on longitudinal observation of subjects during tool use (e.g. [Rolland 99b]). On the other hand, an internet-based method server is being set up. A set of about 40 chunks of scenario-based *RE process models* ([Grosz 97]), reported in [Plihon 98], shall be described. We hope to make this way the CREWS experience of effective and efficient scenario usage in RE open to a large public of researchers and industrials practising scenarios in their everyday lives.

Acknowledgement : Thanks are due to the CREWS leaders and co-workers. The CREWS project involves four partner teams from European Universities, namely : RWTH, Aachen, Germany, the City Univeristy, London, Great Britain, the University Notre-Dame de la Paix, Namur, Belgium, and the University of Paris 1 - Sorbonne, Paris, France. All CREWS reports are available on-line at the CREWS official web-site :

http://SunSITE.Informatik.RWTH-Aachen.DE/CREWS/

5. References

[Antòn 96] A.I. Antòn, *Goal Based Requirements Analysis*. Proceedings of the 2nd International Conference on RE IC RE'96, pp. 136-144, 1996.

[BenAchour 98] C. Ben Achour, C. Rolland, C. Souveyet, *A Proposal for Improving the Quality of Scenario Collections*. Proceedeings of the Fourth International Workshop on RE : Foundations of Software Quality, REFSQ'98, Pisa, Italy Presses Universitaires de Namur, (eds, E. Dubois, A.L. Opdhal, K. Pohl), pp.29-42, 1998.

[BenAchour 98a] C. Ben Achour, *Guiding Scenario Authoring*. Proceedings of the 8th European-Japanese Conference on Information Modelling and Knowledge Bases, Ellivuori, Finland, pp.181-200, 1998.

[BenAchour 99] C. Ben Achour, Extraction des Besoins par Analyse de Scénarios Textuels. Thèse de Doctorat de l'université Paris 6 - Pierre et Marie Curie, 1999.

[BenAchour 99a] C. Ben Achour, C. Rolland, N.AM. Maiden, C. Souveyet, *Guiding Scenario Authoring : Results of an Empirical Study*. To appear in Proceedings of RE'99, fourth IEEE International Symposium on RE, 7-11 June 1999, University of Limerick, Ireland, 1999.

[Bubenko 94] J. Bubenko, C. Rolland, P. Loucopoulos, V De Antonellis, *Facilitating 'Fuzzy to Formal' Requirements Modelling*. IEEE 1st Conference on RE, ICRE'94 pp. 154-158, 1994.

[Carroll 95] J. M. Carroll, M. B. Rosson, *Narrowing the Specification Implementation Gap in Scenario-Based Design*. In Scenario-Based Design: Envisioning Work and Technology in System Development, J.M. Carroll (ed.), pp.247-278, 1995.

[Cockburn 96] A. Cockburn, *Structuring Use Cases with Goals*. Technical report, Human and Technology, http ://members.aol.com/acockburn/papers/ usecase.htm, 1996.

[Dano 97] B. Dano, H. Briand, F. Barbier, *A Use Case Driven RE Process*. Third IEEE International Symposium On RE RE'97, Antapolis, Maryland,

[Dardenne 93] A. Dardenne, A. van Lamsweerde, S. Fickas, *Goal Directed requirements Acquisition*. Science of Computer Programming, 20(1-2), pp.3-50, April 1993.

[Dubois 99] E. Dubois, P. Heymans, *Scenario-Based Techniques for Supporting the Elaboration and the Validation of Formal Requirements*. To appear in RE Journal, 1999.

[Elektra 97] ELEKTRA consortium*, Esprit Program 7.1, Technologies for business processes, best business practice pilots*. Elektra : Electrical Enterprise Knowledge For Transforming Applications. (Proj. N°22927) http ://www.singular.gr/elektra, January 1997 to June 1999.

[Erickson 95] T. Erickson, *Notes on Design Practice: Stories and Prototypes as Catalysts for Communication.* In Scenario-Based Design: Envisioning Work and Technology in System Development, J.M. Carroll (ed.), 1995.

[Firesmith 94] D. G. Firesmith, *Modeling the Dynamic Behaviour of Systems, Mechanisms, and Classes with Scenarios.* In Software DevCon '94, SIGS Publications, pp.73-82,1994.

[Glinz 95] M. Glinz, *An Integrated Formal Model of Scenarios Based on Statecharts*. Lecture Notes in Computer Science, pp.254-271, 1995.

[Grosz 97] G. Grosz, C. Rolland, S. Schwer, S. Souveyet, V. Plihon, S. Si-Saïd, C. Ben Achour, C. Gnaho, *Modelling and Engineering the RE Process : an Overview of the NATURE Approach*. RE Journal, N°2, pp.115-131, Springer-Verlag, 1997.

[Gutttag 93] J.V. Guttag and J.J. Horning, LARCH : Language and Tools for Formal Specification, Springer-verlag, 1993.

[Harel 87] D. Harel, *Statecharts: a Visual Formalism for Complex Systems.* Sci. Computer Program, 8, pp. 231-274, 1987.

[Haumer 98] P. Haumer, K. Pohl, K. Weidenhaupt, *Requirements Elicitation and Validation with Real World Scenes*. IEEE Transactions on Software Engineering, Special Issue on Scenario Management, M. Jarke, R. Kurki-Suonio (eds.), Vol.24, N°12, pp.11036-1054, 1998.

[Holbrook 90] C. H. Holbrrok III, *A Scenario-Based Methodology for Conducting Requirement Elicitation.* ACM SIGSOFT Software Engineering Notes, Vol.15 N°1, pp.95-104, 1990.

[Hsia 94] P. Hsia, J. Samuel, J. Gao, D. Kung, Y. Toyoshima, C. Chen, *Formal Approach to Scenario Analysis*. IEEE Software, pp.33-41, 1994.

[Jacobson 92] I. Jacobson, M. Christerson, P. Jonsson, G. Oevergaard, <u>Object Oriented Software Engineering: a Use Case Driven Approach</u>, Addison-Wesley Publishing Company, 1992.

[Jacobson 95] I. Jacobson, *The Use Case Construct in Object-Oriented Software Engineering.* In <u>Scenario-Based Design: Envisioning Work and Technology in System Development</u>, J.M. Carroll (ed.), pp.309-336, 1995.

[Jarke 96] M. Jarke, *CREWS : Cooperative RE With Scenarios - Project Summary.* http://SunSITE.Informatik.RWTH-Aachen.DE/CREWS/crews-sum.htm

[Jarke 97] M. Jarke, K. Pohl, P. Haumer, K. Weidenhaupt, E. Dubois, P. Heymans, C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N. A.M. Maiden, S. Minocha, *Scenario Use in European Software Organizations – Results From Site Visits and Questionnaires*. CREWS deliverable N°97-10, http:\\SUNSITE.informatik.rwth-aachen.de\CREWS\, 1997.

[Jarke 99] M. Jarke*, CREWS : Towards Systematic Usage of Scenarios, Use Cases and Scenes*. WI (Wirtschaftsinformatik) 99, Saarbrücken, Springer Aktuell , March 1999.

[Johnson 95] J. Johnson, *Chaos : the Dollar Drain of IT project Failures*. Application Development Trends, pp.41-47, January 1995.

[Jones 90] C.B. Jones, <u>Systematic Software Development Using VDM</u>. 2nd edition, Prentice Hall, 1990

[Kyng 95] Morten Kyng, *Creating Contexts for Design*. In Scenario-Based Design: Envisioning Work and Technology in System Development, J.M. Carroll (ed.), pp.85-107, 1995.

[Lalioti 95] V. Lalioti and B. Theodoulidis, *Use of Scenarios for Validation of Conceptual Specification*. Proceedings of the Sixth Workshop on the Next Generation of CASE Tools, Jyvaskyla, Finland, June 1995.

[Leite 97] J.C.S. do Prado Leite, G. Rossi, F. Balaguer, A. Maiorana, G. Kaplan, G. Hadad and A. Oliveros, *Enhancing a Requirements Baseline with Scenarios*. In Third IEEE International Symposium On RE RE'97, Antapolis, Maryland, IEEE Computer Society Press, pp. 44-53, 1997.

[Loucopoulos 94] P. Loucopoulos, *The F3 (From Fuzzy to Formal) View on RE*. Ingénierie des Systèmes d'Information, Vol.2, N° 6, pp. 639-655, 1994.

[Nardi 92] B. A. Nardi, *The Use of Scenarios in Design*. SIGCHI Bulletin, Vol.24 N°4, pp.11-13, 1992.

[Olle 92] T. W. Olle, J. Hagelstein, I.G. MacDonald, C. Rolland, H.G. Sol, F.J.M. Van Assche, A.A. Verrijn-Stuart, Information Systems Methodology : a Framework for Understanding. Addison-Wesley Publishing Company, Second Edition, 1992.

[Opdhal 98] A. L. Opdhal, K. Pohl, *Workshop Summary REFSQ'98*. Proceedeings of the Fourth International Workshop on RE : Foundations of Software Quality, REFSQ'98, Pisa, Italy Presses Universitaires de Namur, (eds, E. Dubois, A.L. Opdhal, K. Pohl), pp.1-11, 1998.

[Pohl 97] K. Pohl, P. Haumer, *Modelling Contextual Information about Scenarios*. Proceedings of the Third International Workshop on RE: Foundations of Software Quality REFSQ'97, Barcelona, pp.187-204, June 1997.

[Plihon 98] V. Plihon, J. Ralyté, A. Benjamen, N.A.M. Maiden, A. Sutcliffe, E. Dubois, P. Heymans, *A Reuse-Oriented Approach for the Construction of Scenario Based Methods*. Proceedings of the International Software Process Association's 5th International Conference on Software Process (ICSP'98), Chicago, Illinois, USA, 14-17 June 1998.

[Potts 94] C. Potts, K. Takahashi, A. I. Antòn. *Inquiry-based Requirements Analysis.* IEEE Software, Vol.11, N°2, pp.21-32, 1994.

[Rawsthorne 96] D. A. Rawsthorne, *Capturing functional Requirements through Object Interactions.* Proceedings of International Conference on RE, ICRE '96, IEEE, pp.60-67, 1996.

[Rolland 98] C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N. A.M. Maiden, Matthias Jarke, P. Haumer, K. Pohl, E. Dubois, P. Heymans, *A Proposal for a Scenario Classification Framework.* RE, Vol.3, N°1, pp.23-47, 1998.

[Rolland 98a] C. Rolland, C. Souveyet, C. Ben Achour, *Guiding Goal Modeling Using Scenarios*. IEEE Transactions on Software Engineering, Special Issue on Scenario Management, Vol.24, N°12, pp.1055-1071, 1998.

[Rolland 98b] C. Rolland, C. Ben Achour, *Guiding the Construction of Textual Use Case Specifications*. Data & Knowledge Engineering, Vol.25, N°1, pp.125-150, 1998.

[Rolland 99] C. Rolland, N. Prakash, A. Benjamen, *A Multi-Model View of Process Modelling*. Submitted to RE Journal, 1999.

[Rolland 99a] Rolland, G. Grosz, R. Kla, *Experience With Goal-Scenario Coupling In RE*. To appear in Proceedings of RE'99, fourth IEEE International Symposium on RE, 7-11 June 1999, University of Limerick, Ireland, 1999.

[Rolland 99b] C. Rolland et al, *Report on the CREWS C2 Evaluations*. Internal report N° 99-18, ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-99-18.ps.gz, 1999.

[Rubin 92] K. S. Rubin, A. Goldberg, *Object Behaviour Analysis.* Communications of the ACM, Vol.35, N°9, pp.48-62, 1992.

[Rumbaugh 91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object-Oriented Modeling and Design. Prentice Hall, 1991.

[Rumbaugh 96] J. Rumbaugh, G. Booch, Unified Method, Notation Summary. Version 0.8, Rational Software Corporation, 1996.

[Somé 96] S. Somé, R. Dssouli, J. Vaucher, *Toward an Automation of RE using Scenarios.* Journal of Computing and Information, Special issue: ICCI'96, 8th International Conference of Computing and Information, Waterloo, Canada, Vol.2, N°1, pp.1110-1132, 1996.

[Spivey 92] J.M. Spivey, The Z notation – A reference manual. 2nd edition, Prentice Hall, 1992.

[Sutcliffe 98] A.G. Sutcliffe, N.A.M. Maiden, S. Minocha, D. Manuel, *Supporting Scenario-based RE.* Transaction of Software Engineering: Special Issue on Scenario Management, Vol. 24, No. 12, December 1998.

[Tawbi 99a] M. Tawbi, F. Velez, C. Ben Achour, *Guiding the Process of Requirements Elicitation Through Scenario Analysis : Results of an Empirical Study*, Proceedings of the First International Workshop on the Requirements Engineering Process - Innovative Techniques, Models, Tools to support the RE Process, Florence, Italy, September 1999.

[VanLamsweerde 95] A. Van Lamsweerde, R. Dairmont, P. Massonet, *Goal Directed Elaboration of Requirements for a Meeting Scheduler : Problems and Lessons Learnt*, in Proceedings Of RE'95 – 2nd Int. Symp. On RE, York, IEEE, 1995, pp 194 –204.

[Weidenhaupt 98] K. Weidenhaupt, K. Pohl, M. Jarke, P. Haumer, CREWS Team. *Scenario Usage in System Development: A Report on Current Practice.* IEEE Software, March, 1998 and International RE Conference (ICRE'98), Colorado Springs, Colorado, USA, April 6-10, 1998.

[Wood 94] D.P. Wood, M. G. Christel, S. M. Stevens, *A Multimedia Approach to Requirements Capture and Modelling.* Proceedings of the International Conference on RE, ICRE'94, Colorado Springs, USA, 1994.

[Wright 92] Peter Wright, *What's in a Scenario.* SIGCHI Bulletin, Vol.24, N°4, October 1992.

[Young 87] M. R. Young, P. B. Barnard, *The Use of Scenarios in Human-Computer Interaction Research: Turbocharging the Tortoise of Cumulative Science*. CHI + GI'87, Human Factors in Computing Systems and Graphics Interface, Toronto, 1987.

[Yu 94] E. Yu, J. Mylopoulos, *Using Goals, Rules and Methods to Support Reasoning in Business Process Reengineering*. 27th Hawaii International Conference System Sciences, Maui, Hawaii, January 4-7, Vol. IV pp. 234-243, 1994.