Crews Report 98-33

Guiding Requirement Engineering with a Process Map

Mustapha Tawbi, Carine Souveyet,

email {tawbi, souveyet}@univ-paris1.fr
Centre de Recherche en Informatique (CRI), University of Paris1 Panthéon Sorbonne,
90 rue de Tolbiac 75013 Paris, France, tel + 33 1 40 77 46 34 (46 04).

Proceedings of MFPE'99 : 2nd International Workshop on the Many Facets of Process Engineering, Gammarth, Tunisia, 12-14 May 1999

Guiding Requirement Engineering with a Process Map

Mustapha Tawbi, Carine Souveyet

email {tawbi, souveyet}@univ-paris1.fr
Centre de Recherche en Informatique (CRI), University of Paris1 Panthéon Sorbonne,
90 rue de Tolbiac 75013 Paris, France, tel + 33 1 40 77 46 34 (46 04).

Abstract

The approach CREWS-L'Ecritoire developed within the CREWS[1] project tightly couples goal modelling and scenario authoring to elicit system requirements. The paper focuses on the process aspect of the approach. It presents the process model as composed of a map and associated guidelines. The map is a directed graph of intentions and strategies to flow from one intention to another. Intention achievement is supported by guidelines. The process model and its enactment mechanism have been implemented in a software tool called L'ECRITOIRE. The paper presents the process map of the approach CREWS-L'Ecritoire and illustrates its use with an example to elicit the requirements of a system to recycle objects.

KEYWORDS : Requirement Chunk, Process Map, Intention, Strategy, L'ECRITOIRE.

Résumé :

L'approche Crews-L'ECRITOIRE développé au sein du projet CREWS couple la modélisation des buts et l'écriture des scénarios pour élucider les besoins d'un système. Le papier traite l'aspect processus de cette approche. Il présente le modèle de processus sous forme d'une carte à laquelle on associe de directives de navigation. La carte est un graphe composé d'intentions et de stratégies. La sélection d'une stratégie permet de progresser d'une intention vers une autre. La satisfaction d'un intention est supportée par une directive. Le modèle de processus et le mécanisme d'exécution associé ont été réalisé dans le logiciel nommé « L'ECRITOIRE ». L'article présente la carte de processus de l'approche CREWS-L'Ecritoire et illustre son utilisation avec un exemple d'élucidation de besoins pour un système de machine à recycler.

Mots Clés : fragment de besoin, carte de processus, Intention, Stratégie, L'ECRITOIRE.

1. INTRODUCTION

Scenario have recently gained attention in the field of Requirement Engineering (RE) [Rolland 94]. A *scenario* is 'a possible behaviour limited to a set of purposeful interactions taking place among several agents' [Caroll 95,Mack 95]. It describes a desirable functionality of a system under design, and thus, help identifying requirements.

Goal modeling is another alternative way to facilitate requirements elicitation [Potts 97]. Our experience is that it is difficult for domain experts to deal with the fuzzy concept of a goal [Bubenko 94, Karadasis 98, Loucopoulos 97 , Rolland 97]. It is often assumed that systems are constructed with some goals in mind [Davis 93]. However, practical experiences [Anton 96, ELEKTRA 97] show that goals are not given and therefore, the question as to where they originate from [Anton 96]acquires importance. In addition, enterprise goals which initiate the goal discovery process do not reflect the actual situation but an idealised environmental one. Therefore, proceeding from this may lead to ineffective requirements. Thus, goal discovery is rarely an easy task.

In the ESPRIT CREWS [2] project, we propose to do this by *combining goal driven approaches* for Requirements Engineering (RE) *with the use of scenarios*. The total solution is in two parts. First, for a goal, scenarios are authored by the scenario author. Thereafter, the authored scenario is explored to yield

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

goals which, in turn, cause new scenarios to be authored and so on. These two main activities had been dealt in [Rolland 98a] for the authoring aspect and in [Rolland 98b]for scenario exploration to discover goals.

The process which combines goal modelling and scenario authoring is a complex one that we want to support by a map of possible routes associated to guidelines. The map guides the user in every phase of the process by indicating what to achieve next ,whereas guidelines helps in telling how to proceed to achieve it. Guidelines are implemented in the software tool L'ECRITOIRE.

The paper is organised as follows. In section 2 we present the notion of requirement chunk and the hierarchy of RCs. In section 3 we present a view of the CREWS-L'Ecritoire map. Section 4 deals with an case study example showing how we use L'ECRITOIRE. to guide the process elicit requirement. We conclude in section 5.

2. THE NOTION OF A REQUIREMENT CHUNK

At the core of our approach is the Requirement Chunk (RC), defined as the pair <goal, scenario> where G is a goal and Sc is a scenario. Since a goal is intentional and a scenario is operational in nature, a requirement chunk is a possible way in which the goal can be achieved[Rolland 98b]. Requirement chunks can be assembled together either through *composition* and *alternative* relationships or through *refinement* relationships. The former lead to AND and OR structure of RCs whereas the latter leads to the organisation of the RCs as a hierarchy of chunks at different levels of abstraction. Figure 1 shows the RC notion using OMT notation[Rumbaugh 91].

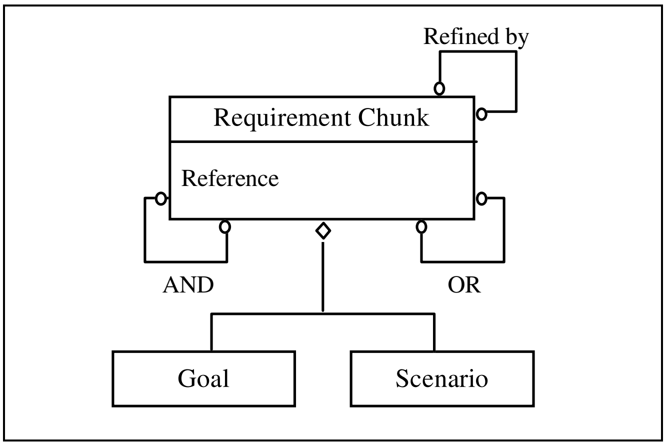

Figure 1:the requirement chunk notion

The three types of relationships among requirement chunks lead to a hierarchical organisation of RCs onto three levels of abstraction , *the behavioural, the functional and the physical* level.

The aim of the *behavioural level* is to identify the services that a system should provide to an organisation and their rationale. A *behavioural chunk* captures a *design alternative* defined by a *design goal* and a *service scenario*. A *design goal* expresses one possible manner of fulfilling the business goal. For example, the design goal ' *Provide paper recycling facilities to our customers with card based machine'* is one possible way of satisfying the business goal. A *service scenario* describes the flow of services among agents (one being the system itself) which are felt necessary to fulfil the design goal. An action of a service scenario is a service such as ' *The customer gets a card from the super market* ' whereas the entire scenario describes the services architecture associated with the design goal. At the *behavioural* level, it is of major importance to explore as many design alternatives as possible i.e. to visualise the various alternative ways by which a system can help an organisation to achieve one of its objectives.

At the *system functional level* the focus is on the interactions between the system and its users. These interactions are required to achieve the services assigned to the system. A *system functional RC* captures one way of providing a service . It couples a *service goal* and a *system interaction scenario*. A *service goal* expresses a manner of providing a service. The associated *system interaction scenario* describes a flow of interactions between the system and its users which allows the users to fulfil the service goal.

The *system physical level* focuses on what the system needs to perform the interactions selected at the system functional level. The *'what'* is expressed in terms of system internal actions that involve system objects but may require external objects such as other systems. System interactions are refined in system internal chunks and new ones are added. A *system physical RC* details one possible way in which the system may internally perform an interaction identified in a system interaction scenario at the previous level. It combines a *system goal* and a *system internal scenario*. A *system goal* expresses a manner to perform an action identified in a system interaction scenario.

3. THE PROCESS MAP

The requirement chunk is the central part of the goal discovery process (Figure 2). This process consists of two phases :

(1) *scenario authoring*
(2) *goal discovery*

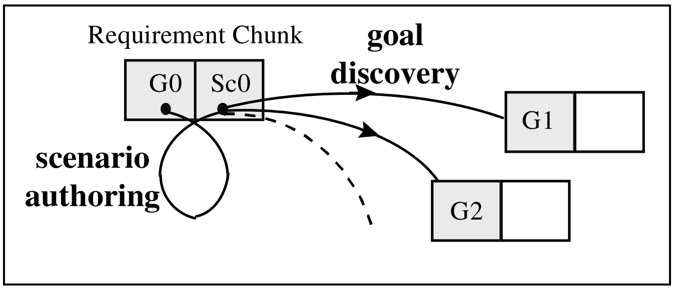

Figure 2 : CREWS-L'Ecritoire process view

Both are supported by strategies, (1) authoring strategies and (2) discovery strategies. Goal discovery and scenario authoring are complementary activities. Once a goal is discovered, scenario authoring can be done, followed by goal discovery through an analysis of the scenario and so on.

For a given goal, a scenario is authored as a possible concretisation of the goal. We assume scenarios to be textual and use authoring strategies to provide style and contents guidelines as well as linguistic devices for analysis, disambiguation and completion. The linguistic devices are based on a case grammar and case patterns [Rolland 98a].

Discovery strategies are of three types to respectively help discovering *ANDed* , *ORed* and *Refined* goals given a requirement chunk RC. These goal-discovery/scenario-authoring sequence is repeated to incrementally populate the requirement chunks hierarchy

The global process of the approach is described by the m*ap* shown in Figure 3. A *map* is a process model in which a non-deterministic ordering of intentions and strategies has been included. It is a labeled directed graph with intentions as nodes and strategies as edges between intentions [Rolland 99]. We assume development processes to be intention-oriented. At any moment, the application engineer has an *intention* that he/she wants to fulfil. To take this characteristic into account the map identifies the set of intentions that have to be achieved in order to solve the problem at hand.

Intention is expressed as a natural language statement comprising a verb and several parameters, where each parameter plays a different role with respect to the verb[Prat 97]. The examples below introduce the parameters useful in this paper (For more details see [Rolland 98a])

We consider here three types of parameters: *Target*, *Source*, and *Way*.

The *Target* designates entities affected by the intention. We distinguish two types of targets, *Objects* and *Results*. An *Object* is supposed to exist before the goal is achieved. For example in the goal statement :

Conceptualize $_{\text{verb}}$ *Scenario* $_{\text{object}}$ *manually*

The target '*Scenario*' is an object because it exists even before '*Conceptualise*' is achieved. *Results* can be of two kinds (a) entities which do not exist before the goal is achieved (b) abstract entities which exist but are made concrete as a result of goal achievement

A *strategy* is an approach, a manner to achieve an intention. The strategy, as part of the section <Ii,Ij,Sij> characterizes the flow from Ii to Ij and the way Ij can be achieved.

As shown in Figure 3, there might be several flows from an intention to another intention , each corresponding to a specific strategy. In this sense the map offers *multi-thread flows*. Finally, the map can include reflexive flows (see ' case based discovery strategy ' in the Figure 3)

The directed nature of the graph shows which intentions can follow which one. A map consists of a number of *sections* each of which is a section < Ii, Ij, Si,j > where Ii and Ij are two intentions and Si,j is a strategy. There are two distinct intentions called *Start* and *Stop* respectively that represent the intentions to start navigating in the map and to stop doing so. Thus, it can be seen that there are a number of paths in the graph from *Start* to *Stop*.

The map to follow by the approach is shown in Figure 3

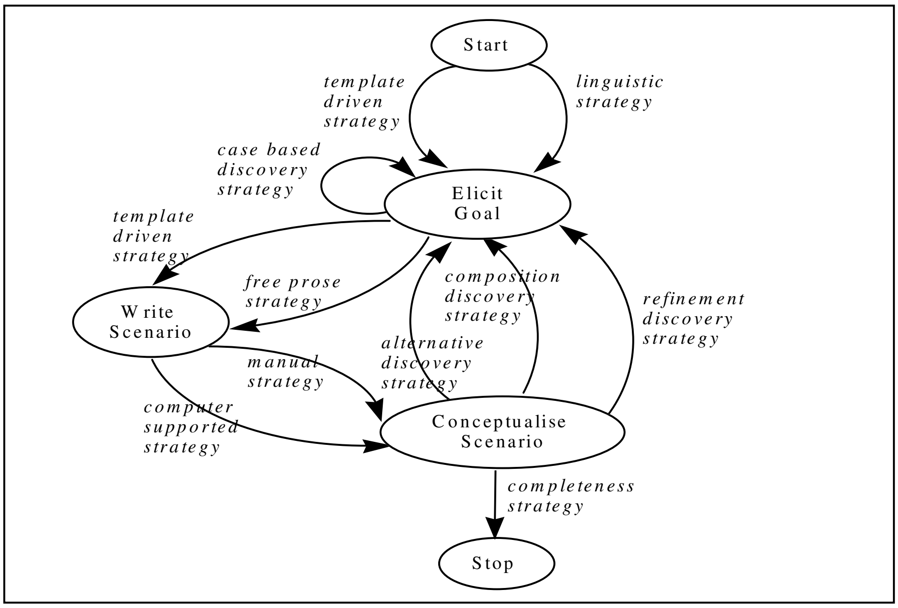

Figure 3 :The Process Map

As it can be seen, there are five intentions *Start, Elicit Gaol, Write Scenario, Conceptualise Goal ,and Stop, and e*leven strategies are offered to fulfil these intentions.

As a consequence, the navigation through the map is done through 11 *sections* :

1. < *Start, Elicit goal,* ***template driven strategy*** >
2. < *Start, Elicit goal,* ***linguistic strategy*** >
3. < *Elicit goal, Elicit goal,* ***case based discovery strategy*** >
4. < *Elicit goal, Write scenario,* ***Free prose strategy*** >
5. < *Elicit goal, Write scenario,* ***template driven strategy*** >
6. < *Write scenario, Conceptualise scenario,* ***manual strategy*** >
7. < *Write scenario, Conceptualise scenario,* ***computer supported strategy*** >
8. < *Conceptualise scenario, Elicit Goal,* ***refinement discovery strategy*** >
9. < *Conceptualise scenario, Elicit Goal,* ***composition discovery strategy*** >
10. < *Conceptualise scenario, Elicit Goal,* ***alternative discovery strategy*** >
11. < *Conceptualise scenario, Stop,* ***completeness strategy*** >

Each of these sections is associated with a guideline which provide advises on how to achieve the intention.

4. A scenario of use of the L'ECRITOIRE software tool

The CREWS-L'ECRITOIRE map is implemented in a CASE tool called L'ECRITOIRE [Tawbi 98,Souveyet 98], L'ECRITOIRE guides the user by indicating him/her at any moment his/her position in the map, the next intentions to achieve and the set of strategies that he can apply to achieve them.

The user starts always from the intention *'Start', then he/she* chooses a strategy from the list of strategies offered by the map to move to the next intention. Once the user selects a strategy, the tool enacts the corresponding guideline. At the end of enactment, the user reaches the target intention and so on until reaching the intention 'Stop'.

In this section we present an example describing the process to elicit the set of requirements needed in order to develop the software controlling a recycling machine for customers in a super market. We show how to use L'ECRITOIRE to get the guidance and help during the requirements elicitation process.

Figure 4 shows in bold the path chosen by the user of the tool that will be exemplified in the rest of the paper .

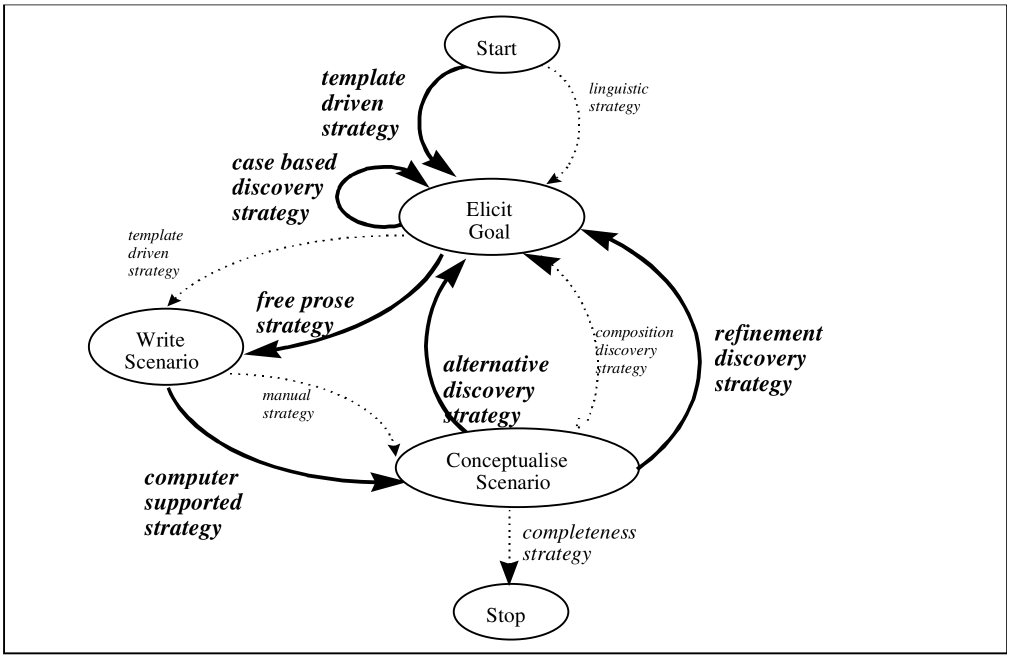

Figure 4 : Path followed by the user during the example process

Obviously, this path is in fact constructed dynamically during the elicitation process in the normal course of actions.

5.1 Illustrating the *'template driven strategy'*

The user of the tool called the requirement chunk author (RCA), starts by eliciting a goal at the behavioural level which is **'Provide Recycling Facilities'.** To do this, he chooses the section '< *Start, Elicit goal,* **template driven strategy**>'. Once this strategy is chosen, L'ECRITOIRE enacts the interface 'Elicit Goal' shown in Figure 5. The RCA uses this interface to rephrase the goal according to the suggested template, to precise the manner to fulfil it and the type of the goal. By pushing the button 'OK' the RCA achieves the intention 'Elicit goal' and L'ECRITOIRE invites him to move to the next intention in the map.

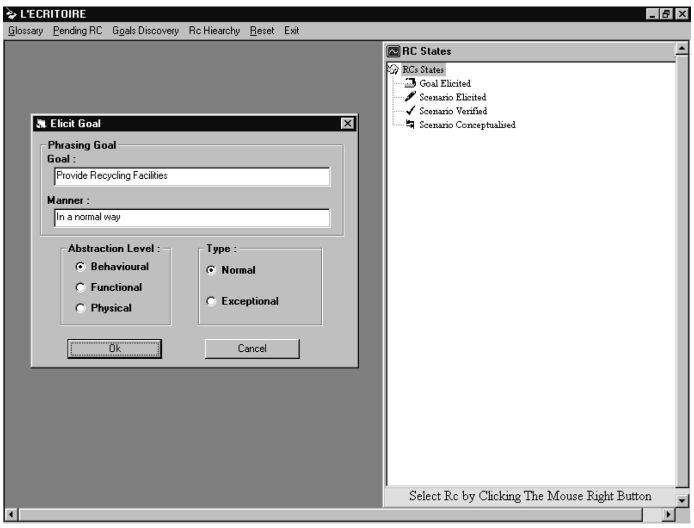

Figure 5 : The interface 'Elicit Goal'

L'ECRITOIRE uses the window 'RC States' on the left of Figure 5 to propose to the RCA the next intentions to achieve. This window indicates to the RCA his position with respect to the initial map. The goal written in the previous step is the leaf of the branch 'Elicited goals' in tree 'RC states'.

5.2 Illustrating the '*case based discovery strategy'*

To progress the RCA chooses one goal from the tree. Once he chooses it, L'ECRITOIRE indicates him that he can apply now one of three strategies ***case based discovery strategy, free prose strategy*** *or* ***template driven strategy.***

As shown by the map, these three strategies corresponds to section '3','4', and '5' in the list of 11 sections presented above.

Assumes that the RCA chooses the section *'< Elicit goal, Elicit goal,* ***case based discovery strategy****>',* *L'ECRITOIRE then enacts the interface ' Elicit Goal by Goal structure Analysis'* (Figure 6).

The '*case based discovery strategy'* uses the goal structure [Rolland 98a,Prat 97] in order to generates other goals which are alternatives to the goal at hand.

The goal structure associates a verb to a set of parameters, each parameter playing a different role with respect to the verb.
The parameters are source, target, beneficiary, destination, means and manner. Every goal must have a verb and a target whereas other parameters are optional.

the structure of the goal '**Provide Recycling Facilities'** is the following :

verb	source	target	beneficiary	destination	means	manner
provide	' '	Recycling Facilities	' '	' '	' '	' '

Based on this structure, the RCA is asked to provide alternative values for the target parameter. He proposes '*Paper Recycling Facilities', 'Bottles Recycling Facilities', 'Boxes Recycling Facilities, and 'Boxes and Paper Recycling Facilities* 'as alternatives to the initial **target**. He proposes '*our customers', and 'all customers'* as possible alternative **destinations.** And he proposes*' with card based machine' and 'with money return machine'* as possible alternative **means.**

The tool generates automatically a list of new goals by computing all the possible combination between alternative values. The RCA is asked then to select the set of the most relevant ones and to eliminate the non relevant ones.

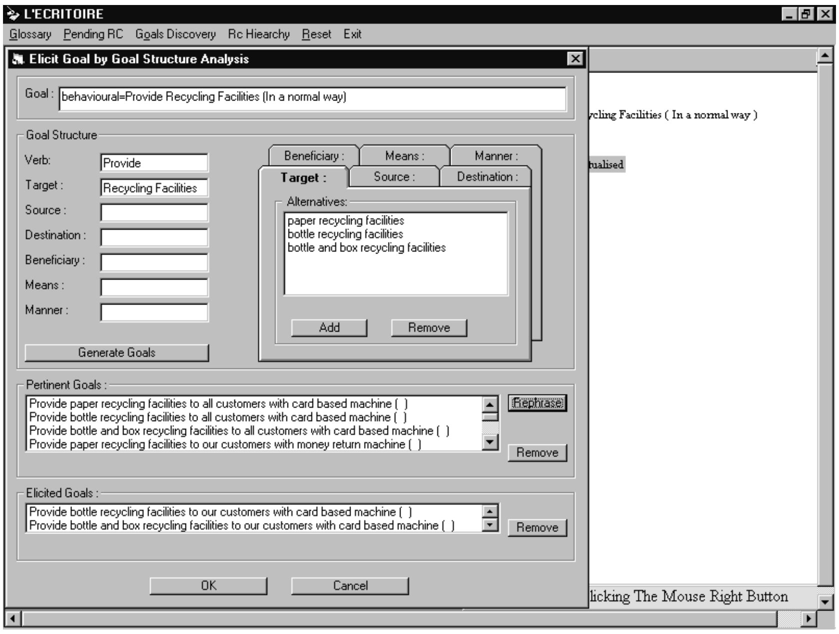

Figure 6: The interface 'Elicit Goal by Goal Structure Analysis'

In this example, we suppose that the RCA keeps the following three goals as they are the most relevant according to his point of view :

2. Provide paper recycling facilities to our customers with card based machine
3.Provide bottle recycling facilities to our customers with card based machine
4.Provide bottle and box recycling facilities to our customers with card based machine

These goals are at the behavioural level and they are linked with an 'OR' relationship since they suggest alternatives to '**Provide Recycling Facilities'** .

5.3 Illustrating the '*free prose strategy'*

In order to progress, L'ECRITOIRE indicates to the RCA that he has now four goals having the state elicited and therefore he can apply on each one of them one of the three strategies ***case based discovery strategy, free prose strategy*** *or* ***template driven strategy.***

Suppose that the RCA chooses the goal '4' as his alternative to design the desired recycling system. And he selects the section '*< Elicit goal, Write scenario,* ***free prose strategy****>'* as the next one.

In this case, L'ECRITOIRE enacts the interface 'Write Scenario' and the RCA is asked to write a scenario describing the way to achieve this goal. 'Style and content guidelines' are offered to help the RCA during the authoring of the scenario (Figure 7). Style guidelines deal with the wording of the text, whereas content guidelines focuses on the nature of the scenario content.

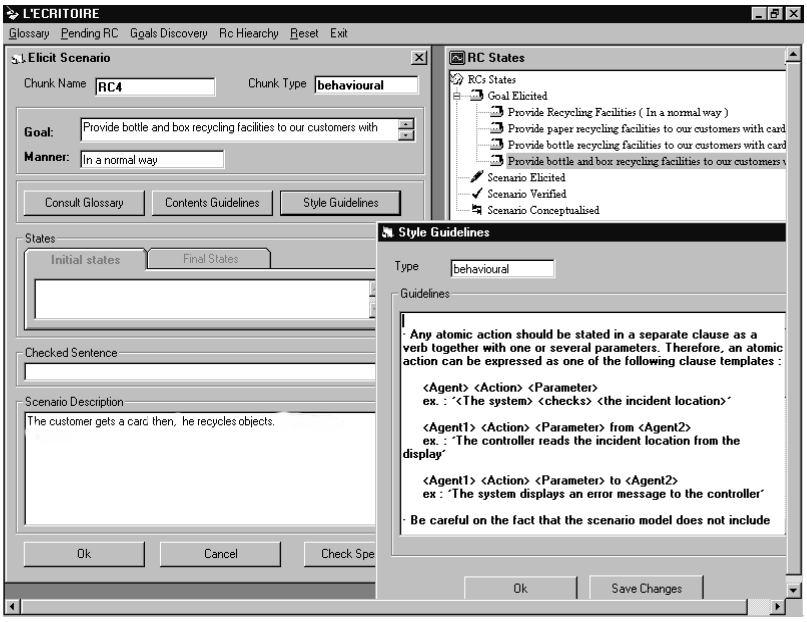

Figure 7: The interface 'Elicit Scenario'

The RCA writes the following scenario :

> 'The customer gets a card then, he recycles objects.'

5.4 Illustrating the *'computer supported strategy'*

Once the scenario is authored, L'ECRITOIRE indicates to the RCA that goal '*Provide bottle and box recycling facilities to our customers with card based machine* ' has now an associated scenario and it proposes to select one of tow sections '*< Write scenario, Conceptualise scenario,* ***manual strategy****>' or '< Write scenario, Conceptualise scenario,* ***computer supported strategy****>'.*

Supposes that the RCA select the section '< *Write scenario, Conceptualise scenario*, ***computer supported strategy***>'.

The fulfilment of the intention 'Conceptualise scenario' requires the application of two sub-strategies ' Verify and clarify Scenario' and 'Map scenario'. The first sub-strategy aims to complete and clarify actions in the initial scenario, whereas the latter aims to map the scenario onto formal description.

the 'Verify and Clarify Scenario' sub-strategy uses linguistic devices to analyse the semantic content of the scenario and to represent it as a collection of instantiated linguistic case patterns [Rolland 98a]. There are two types of semantic patterns, **clause** and **sequence patterns.** The former provide the semantic of atomic actions such as" The customer gets a card from the super market" whereas the latter provide the semantic of complex actions such as " The customer gets a card from the super market then, he recycles object"

A verb is the main concept of a **clause pattern** . A verb is the centre and we attach to it the other parameters of the sentence. We classify verbs in two categories : *action verbs* and *communication verbs.* *Action verbs* care are used in simple actions such as "check " or " validate ". An action verb requires an agent parameter which is the pronoun of the verb and an object parameter which is the subject of the verb. Then, the sentence " the customer recycles objects" will be represented as the instantiated pattern :Action(recycle)[agent : the customer; object : objects].

Communication verbs are used in actions where agents exchange objects which can be physical objects like 'a card' or information objects like 'a message 'or 'a prompt'. For example in the action " The customer gets a card from the super market" 'get' is a communication verb and 'a card' is a physical object. A communication verb needs also two parameters : a source ('the super market') and a destination('the customer'). So" The customer gets a card from the super market" corresponds to the instantiated pattern :Communication (get) [Agent : the customer ; Object : a card ; Source : the super market ; destination : the customer] .

The purpose of **Sequence patterns** is to compose simple actions in complex ones. We classify complex actions in four categories : *Sequence, Condition* **,** *Iteration and Concurrency* [Rolland 98a].

L'ECRITOIRE uses 'Verify and Clarify Scenario' sub-strategy to generates automatically the instantiated patterns corresponding to the initial scenario description . the tool replace missing parameters in an action by a ' ? ' and it asks the user to replace them in the action by the corresponding parameter .
Let us, for example, consider the sentence " the customer gets a card" and its corresponding instantiated pattern

communication(get) [Agent : the customer ; object : a card ; Source : ? ; Destination : the customer]
The RCA is asked to replace every '?' by a term. This leads to the completed sentence :
" The customer gets a card from the super market"

Anaphoric references are detected and replaced by non ambiguous terms. For example in

the action " he recycles objects**"** the anaphoric reference 'he' is detected and the RCA is asked to replace 'his' by a term, 'the customer' and the action is rephrased as
*" the customer r*ecycles objects*" let us say.*

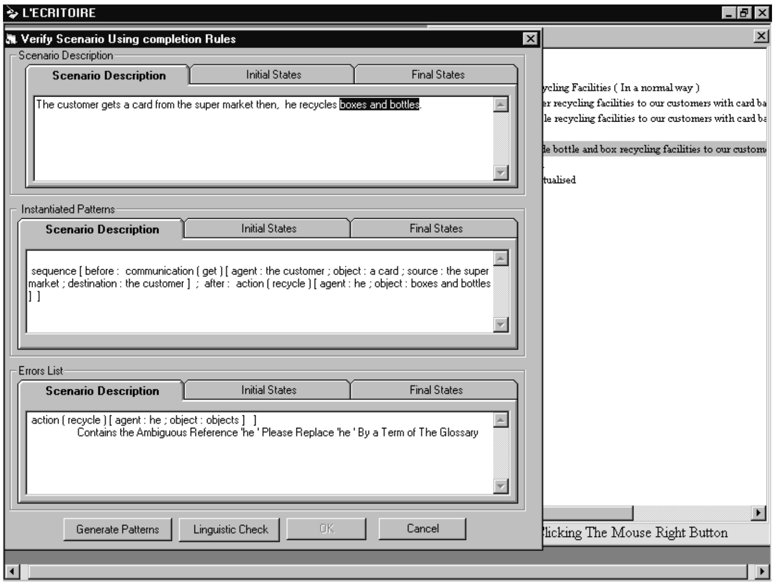

Figure 8: the interface 'Verify Scenario Using Completion rules'

The following description corresponds to the scenario revised :

'The customer gets a card from **the super market** then, **the customer** recycles **boxes and bottles**.'

The ' Map scenario' sub-strategy is performed automatically by the tool (Figure 9). It transforms the initial into the following one :

1. the customer gets a card from the super market 2. the customer recycles boxes and bottles

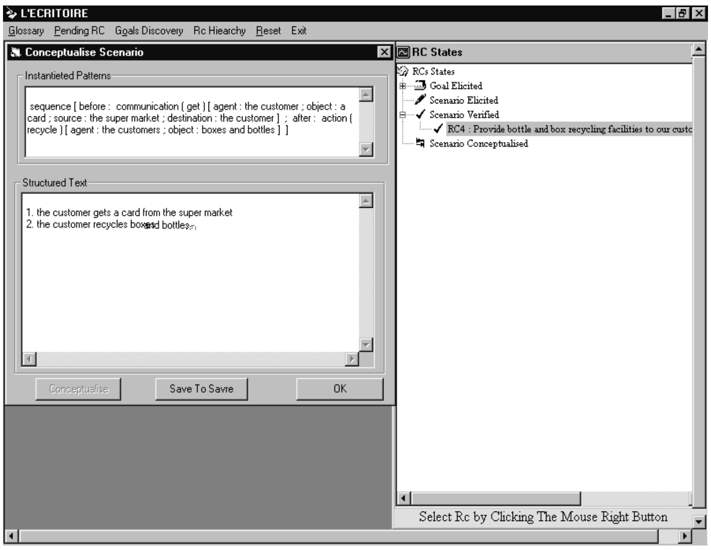

Figure 9: The interface 'Conceptualise Scenario'

5.5 Illustrating the '*refinement discovery strategy'*

Once the intention 'Conceptualise Scenario' is fulfilled, L'ECRITOIRE suggests to the RCA to apply one of the three '**Goal discovery**' strategies on.

Suppose now that he chooses the section '*< Conceptualise scenario, Elicit Goal,* ***refinement discovery strategy*** *>.'*

This strategy looks to every interaction in a scenario at level i as a goal at level i+1. The tool scans every interaction in a scenario and asks the RCA to confirm or infirm the fact that the interaction should be regarded as a goal. In the positive case, the RCA is asked to rephrase the selected action as a goal statement. The discovered goal belongs to a requirement chunk which refines the initial RC. As shown in below, the RCA decides to treat he atomic action " the customer *gets a car from the super market* " at the *behavioural* level as a goal re-named " *Get a card from the super market* " at the functional level .

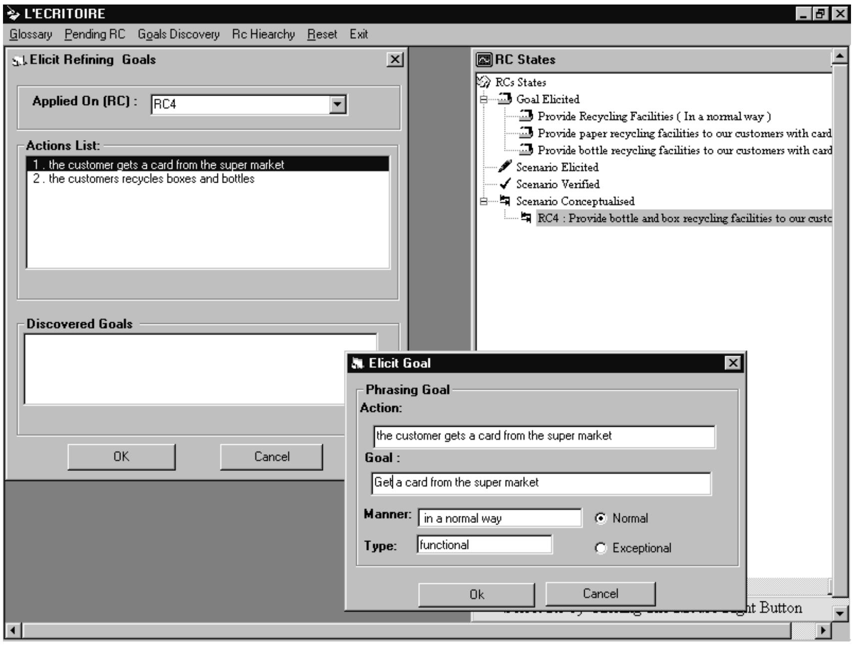

Figure 10: The interface 'Elicit Refining Goals'

By applying the refinement strategy, the RCA elicits the two following goals :

4.1 Get a card from the super market
4.2 Recycle boxes and bottles

'4.1 'and '4.2' are at the functional level. They are linked with an 'Refine' relationship to goal '3' and with an 'AND' relationship to each other.

5.6 Illustrating the 'alternative *discovery strategy'*

The same map on the goal '4.2' is summed up as follows.

Given the initial scenario :

> The customer inserts his card in the RM. The RM checks if the card is valid and then a prompt is given. The customer inputs the bottles and or/the boxes in the RM. If the objects are not blocked, the RM ejects the card and prints a receipt.

The verified and completed scenario (modifications are shown in bold) is the following:

> The customer inserts a card in the RM. The RM **checks the card validity**. If the card is valid then a prompt is given **by the RM to the customer**. The customer inputs the bottles and or/the boxes **in the RM**. If bottles and/or the boxes are not blocked, the RM ejects the card **to the customer** and **the RM** prints a receipt **to the customer**.

And the structured version is :

> 1. the customer inserts a card in the RM
> 2. the RM checks the card validity

3. If the card is valid
Then
4. a prompt is given by the RM to the customer
5. the customer inputs the bottles and/or the boxes in the RM
6. If the bottles and/or the boxes are not blocked
Then
7. the RM ejects the card to the customer
8. the RM prints a receipt to the customer

As explained before, once a scenario is conceptualised, L'ECRITOIRE proposes the three '*goal discovery*' strategies. Assume that the RCA chooses the ***'alternative discovery strategy'*** **.**

This strategy aims to identify ways to achieve a given service goal but in different manners. It builds a graph representing all the possible paths of actions already identified in the scenario of the requirement chunk under investigation. Every path is characterised by *zero* to *n* nested flow conditions. For example, in the scenario associated to the goal '*Recycle boxes and bottle's,* there are two nested flow conditions

1- If the card is valid
2- If the bottles and/or the boxes are not blocked

The **'alternative discovery strategy'** computes all the combinations of negative conditions that should be investigated as possible missing paths. S2 generates two cases in our example :

(1)	*card is not valid*
(2)	*(card is valid) & (the bottles and/or the boxes are blocked)*

These cases help to discover new goals such as *'Recycle boxes and bottles with invalid card'*.

These goals belong to RCs that are related to the initial RC through an OR relationship.

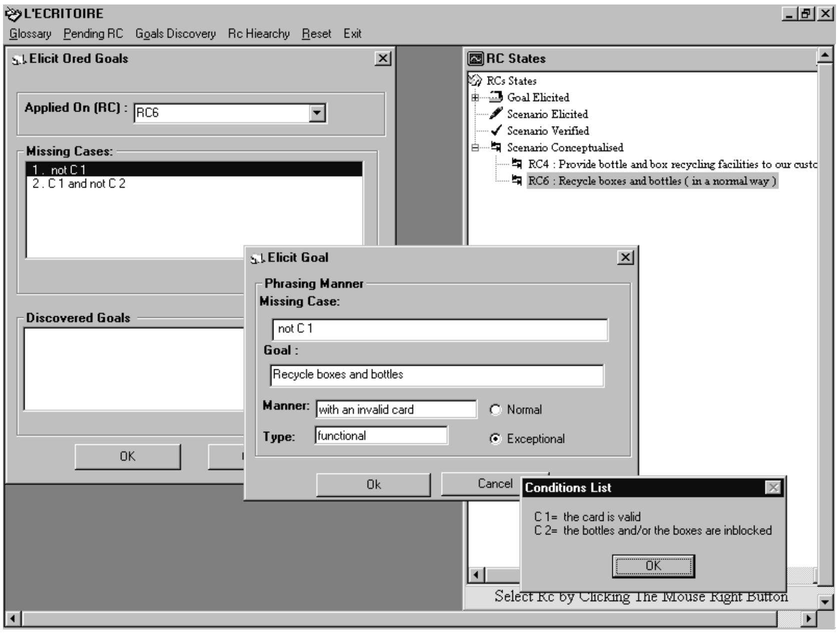

Figure 11: The interface 'Elicit Ored Goals'

By applying this strategy the RCA obtains the two following goals :

4.2(1)Recycle boxes and bottles with invalid card
4.2.(2) Recycle boxes and bottles with a blocked box or bottle.

At this step of the process, the gaols and RCs that have been elicited
are classified in the following :

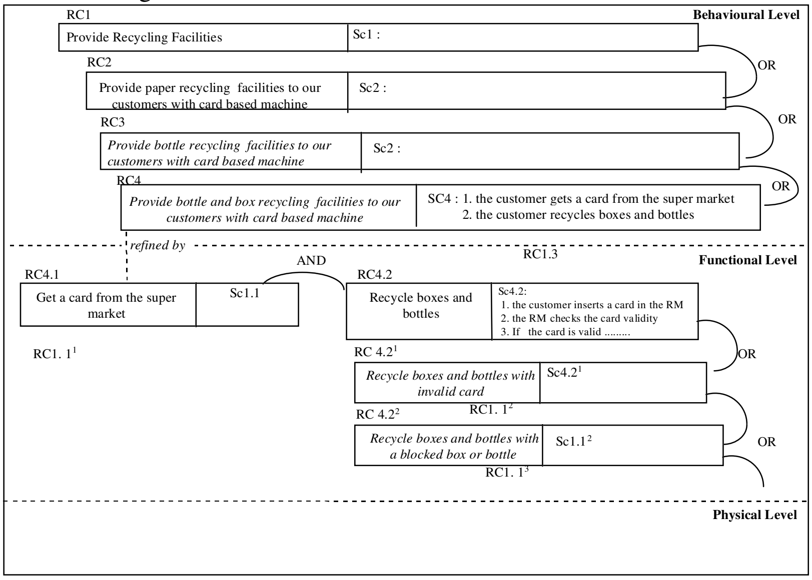

Figure 12 : RCs Hierarchy

The RCA is now invited to start similar process for every of the 'Elicited goal ' and so on. The entire process ends when reaching the intention 'Stop' using the '***completeness strategy'*** . This strategy consists of verifying that all elicited goals are associated to scenarios and properly encapsulated in requirement chunks.

5.Conclusion

We presented the CREWS-L'Ecritoire approach to guide requirements elicitation by combining goal modelling and scenario authoring. The all process is guided dynamically using a map. A map is composed of intentions and strategies, strategies being used to reach intentions. The map was implemented by a CASE tool called L'ECRITOIRE. The paper illustrate one path in the map with the 'Recycling machine' example. The approach has been also validated with examples such as the ATM case study. We are currently working in two directions (a) validating the approach by considering complete cases and diversifying cases and (b) defining new strategies using our goal_scenario based approach. We have at the moment 6 supplementary strategies under investigation. Once these strategies will be finalised, they will be implemented to improve the flexibility of the requirement engineering process .

7.Références

[Anton 96]A.I. Anton, *Goal based requirements analysis*. Proceedings of the 2nd International Conference on Requirements Engineering ICRE'96, pp. 136-144, 1996.

[Bubenko 94] J. Bubenko, C. Rolland, P. Loucopoulos, V De Antonellis, *Facilitating 'fuzzy to formal' requirements modelling*. IEEE 1st Conference on Requirements Enginering, ICRE'94 pp. 154-158, 1994.

[Caroll 95] J.M. Caroll, The Scenario Perspective on System Development, in J.M. Carroll (ed.), Scenario-Based Design: Envisioning Work and Technology in System Development (1995).

[Davis 93] A.M. Davis, *''Software requirements :objects, functions and states''*. Prentice Hall, 1993.

[ELEKTRA 97] ELEKTRA consortium, *Electrical Enterprise Knowledge for Transforming Applications - Athena deliverable : initial requirements for PPC.* ELEKTRA Project Internal Report, 1997.

[Karadasis 98] P. Kardasis, P. Loucopoulos, *Aligning legacy information system to business processes.* Submitted to CAiSE'98, 1998.

[Loucopoulos 97] P. Loucopoulos, V. Kavakli, N. Prekas, *Using the EKD approach, the modelling component.* ELEKTRA project internal report, 1997.

[Mack 95] R.L. Mack, Discussion : Scenarios as Engines of Design, in John M. Carroll (ed.), Scenario-Based Design: Envisioning Work and Technology in System Development (John Wiley and Sons, 1995) 361-387.

[Potts 97]C. Potts, *Fitness for use : the system quality that matters most.* Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'9 , Barcelona, pp. 15-28, June 1997.

[Prat 97] N. Prat, *Goal formalisation and classification for requirements engineering*. Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'97, Barcelona, pp. 145-156, June 1997.

[Rolland 94] C Rolland, G. Grosz , A General Framework for describing the requirement engineering process ", IEEE conference on Systems, Man & Cyberenetics, CSMC'94, San Antonio, Texas 1994.

[Rolland 97] C. Rolland, S. Nurcan, G. Grosz, " *Guiding the participative design process.* Association for Information Systems Americas Conference, Indianapolis, Indiana, pp. 922-924, August, 1997.

[Rolland 98a]C. Rolland, C. Ben Achour, *Guiding the construction of textual use case specifications*. Published in Data and Knowledge Engineering Journal 25 (1998) p125-160.

[Rolland 98b] C. Rolland, Carine Souveyet, Camille Ben Achour : " Guiding Gaol Modelling Using Scenario " , Published in the TSE special issue on Scenario Management 1998.

[Rolland 99] C. Rolland, A. Benjamen , A Multi-Model View of Process Models, submitted to REJ in 1999 (Requirement Engineering Journal) .

[Rumbaugh 91]J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *''Object-oriented modelling and design''*. Prentice Hall, 1991.

[Souveyet 98] C . Souveyet, M. Tawbi : " Process centred Approach for developing tool support of situated methods ", Proceeding of the conference DEXA'98, 9th International Conference and Workshop on. Database and Expert Systems Applications. August1998, Austria, Vienna

[Tawbi 98] M. Tawbi, C. Souveyet, C. Rolland, L'ECRITOIRE a tool to support a goal-scenario based approach to requirements engineering, submitted to IST (Information and Software Technology Journal in 1998.