CREWS Report 98-32

Fourth IEEE International Symposium on Requirements Engineering (RE'99), 7-11 June 1999, University of Limerick, Ireland

Experience With Goal-Scenario Coupling In Requirements Engineering

by

Colette Rolland, Georges Grosz, Régis Kla

Centre de Recherche en Informatique
Université Paris 1 - Sorbonne
90, rue de Tolbiac
75013 Paris - France
{rolland, grosz, kla}@univ-paris1.fr

Experience With Goal-Scenario Coupling In Requirements Engineering

Colette Rolland, Georges Grosz, Régis Kla
Université Paris 1 - Panthéon – Sorbonne, Centre de Recherche en Informatique
90, rue de Tolbiac, 75634 Paris Cedex FRANCE
email : {rolland, grosz, kla}@univ-paris1.fr

Abstract

In the context of Requirements Engineering (RE), both goal-driven and scenario-based approaches have proven useful for eliciting, justifying and validating system requirements. In order to overcome some of the deficiencies and limitations of these approaches when used in isolation, proposals have been made to couple goals and scenarios together. The CREWS[1]-L'Ecritoire approach advocates bi-directional coupling precisely to permit two ways movement between goals and scenarios.

The paper reports an experiment using the CREWS-L'Ecritoire approach on a large scale business re-engineering project conducted for an electricity supply and distribution company. We focussed on a set of issues we found important and of a significance beyond the limits of our brief. We considered three kinds of issue as found in goal-driven approaches, in scenario-based approaches, and those specific to goal-scenario coupling. In this paper, we devote particular attention to the latter and assess the extent to which the goal-scenario coupling helps to resolve the first two.

1. Introduction

Requirements Engineering (RE) is concerned with the elicitation and definition of system requirements. Whereas initial research efforts focused on the requirements definition facet [25, 10, 24, 14] and address *what* questions only, recent attempts have been made to develop approaches that support the requirements elicitation facet. Within these approaches *why* questions can be addressed in addition to the usual *what* questions. It then becomes expected to elicit system requirements better meeting the organisation's goals and needs. *Goal-oriented RE* and *scenario-based RE* are two distinct trends aiming at eliciting requirements from an analysis of the wider context in which the system will operate.

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 [16]. RE is therefore concerned with the elicitation of high-level goals to be achieved by the envisioned system [2, 3, 7], the refinement of these goals [7, 30, 23] and their operationalisation into system requirements specifying how goals should be accomplished by the proposed system [2]. However, practical experience shows that (a) goal discovery is not an easy task [23, 2], (b) application of goal reduction methods [7] to discover component goals of a goal is not as straight-forward as the literature suggests [3, 8, 2] and (c) eliminating uninteresting and spurious goals is necessary and difficult [20].

Independently of goal modelling, an alternative approach to RE, the *scenario-based approach*, has been developed. Scenarios have proved useful in requirements elicitation in a number of ways : to elicit requirements in envisioned situations [19], to help in the discovery of exceptional cases [19, 23, 26], to derive conceptual models [6, 24, 14], to reason about design decisions [4] and so on. The argument is 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 [29]. In addition, since scenarios describe concrete behaviours, they capture real requirements. However, because they deal with examples and illustrations, scenarios are inherently partial and only provide restricted requirements descriptions which need to be generalised to obtain complete requirements.

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 [6, 14, 15, 18] goals are considered as contextual properties of use cases whereas in [5] they are used as a means to structure use cases. The goal scenario combination has been used to operationalise goals [2, 13, 19, 23], to check whether or not the current system usage captured through multimedia scenarios fulfils its expected goals [12], to infer goals specifications from operational scenarios [28] and to discover new goals through scenario analysis [23].

The purpose of this paper is to assess the strengths and weaknesses of an approach to requirements engineering that uses goal-scenario coupling. The CREWS-L'Ecritoire approach [22, 23] developed within the CREWS ESPRIT project uses *a bi-directional coupling* allowing movement from goals to scenarios and vice-versa. The complete 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. In this process, goal discovery and scenario authoring are complementary steps and goals are incrementally discovered by repeating the goal-discovery, scenario-authoring cycle.

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

The assessment is based on a real case study conducted in the context of Business Process Reengineering [11]. The case study is part of the ELEKTRA project [8] aiming at developing and experimenting with an approach to managing change in electricity supply and distribution companies due to deregulation rules issued by the European Community. In the context of business process reengineering, the issues of *why* are critical to understand current processes and identify deficiencies in those processes, to envision new processes and to evaluate the different alternative change scenarios. The case study reported in this paper was performed in an electricity distribution company of 35,000 employees : PPC, Greece. The focus here is on the issues that we found important and that we believed are not specific to our approach. We considered three kinds of issues, those which are found in goal-oriented approaches, those in scenario-based approaches, and those specific to goal-scenario coupling. In this paper, our attention will be, on the one hand, to consider the third kind of issues and on the other hand, to assess the extent to which the goal-scenario coupling helps to resolve the first two kinds of issues.

The CREWS-L'Ecritoire approach is summarised in section 2. Section 3 follows step by step the application of the approach to the ELEKTRA case study and discusses the issues of interest. The examples are not connected together, they aim at illustrating the issues. An extended discussion of the issues, their origin and resolution, and the impact of the goal-scenario coupling on these issues concludes this paper in section 4.

2. An overview of the CREWS l'Ecritoire approach

In this section we first present some of the key concepts and terminology of the CREWS L'Ecritoire approach and then provide an overview of its process.

2.1. Concepts and terminology

A ***Requirement Chunk*** (RC) is a pair <G, Sc> 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 of achieving the goal.

A ***goal*** is defined as "something that some stakeholder hopes to achieve in the future" [17]. In our approach, a goal [21] is expressed as a clause with a main verb and several parameters, where each parameter plays a different role with respect to the verb. A detailed description of the goal structure can be found in [23]. An example of a goal is the following :

Provide $_{\text{verb}}$ (efficiently) $_{\text{quality}}$ (electricity) $_{\text{object}}$ (from PPC producer) $_{\text{source}}$ (to our non eligible customers) $_{\text{beneficiary}}$ (using PPC network) $_{\text{means}}$ (in a normal way) $_{\text{manner}}$

A ***scenario*** is "a possible behaviour limited to a set of purposeful interactions taking place among several agents" [17]. It is composed of one or more *actions*, an *action* being an interaction *from* one agent *to* another. The combination of actions in a scenario describes a unique path. A scenario is characterised by an initial and a final state. An *initial state* attached to a scenario defines a precondition for the scenario to be triggered. A *final state* defines a state reached at the end of the scenario. We distinguish between *normal* and *exceptional* scenarios. The former leads to the achievement of its associated goal whereas the latter fails in goal achievement.

Requirement chunks classification and ***abstraction levels*** : We have introduced three levels of abstraction called *contextual*, *functional*, and *physical*. The contextual level identifies the services that a system should provide to an organisation and their rationale. The functional level focuses on the interactions between the system and its user to achieve the needed services. Finally, the physical level deals with the actual performance of the interactions. Each level corresponds to a type of RC. As a result, we organise the collection of requirements in a three level abstraction hierarchy.

Relationships between requirement chunks: There are three types of relationships among Requirement Chunks (RC) namely, the composition, alternative, and refinement relationships. The first two of these lead to a horizontal AND/OR structure between RCs. These are extensions of conventional AND/OR relationships between goals. AND relationships among RCs link together those chunks that require each other to define a completely functioning system. RCs related through OR relationships represent alternative ways of fulfilling the same goal. The third kind of relationship relates requirement chunks at different levels of abstraction. The refinement relationship establishes a vertical link between requirement chunks.

2.2. The requirements elicitation process

The CREWS-L'Ecritoire process aims at discovering/eliciting requirements through a bi-directional coupling of goals and scenarios allowing movement from goals to scenarios and vice-versa. As each goal is discovered, a scenario is authored for it. In this sense, the goal-scenario coupling is exploited in the forward direction from goals to scenarios. Once a scenario has been authored, it is analysed to yield goals. This leads to goal discovery by moving along the goal-scenario relationship in the reverse direction.

The exact sequence of steps of the process is:

1. Initial Goal Identification
repeat 2. Goal Analysis
3. Scenario Authoring
4. Goal Elicitation Through Scenario Analysis
until all goals have been elicited

It can be seen that goal elicitation and scenario authoring are complementary steps and goals/ requirements are incrementally discovered by repeating the goal-analysis, scenario-authoring, goal-elicitation-through-scenario-analysis cycle. Each of the three steps of the cycle is supported by mechanisms to guide the execution of the step.

The guidance mechanism for goal analysis is based on a linguistic analysis of goal statements. It helps in reformulating a narrative goal statement as a goal template as introduced in the previous section. The mechanism for scenario authoring combines style/content guidelines and linguistic devices. The former advise authors on how to write scenarios whereas the latter provides semi-automatic help to check, correct, conceptualise, and complete a scenario. Finally, for goal elicitation through scenario analysis, we defined enactable rules offering three different goal discovery strategies namely, refinement strategy, composition strategy, and alternative strategy. The first of these discovers goals at a lower level of abstraction than a given goal ; the second discovers goals ANDed to the original one ; the last discovers goals ORed to the original goal.

3. The ELEKTRA case study

In this section, we will apply the CREWS-L'Ecritoire approach to the ELEKTRA case study with a view to identifying how the requirements elicitation issues are addressed and resolved. The specific issues handled in each step of this application are discussed.

3.1 Initial Goal Identification

The approach requires the identification of the highest level goals of the problem at hand. This can be done through a suitable study of initial documents and brain-storming sessions. These goals may be at any of the three levels of abstraction and become the starting point of the requirements elicitation process. In the ELEKTRA case study, we identified one goal at the contextual level, namely, "Run PPC in TPA manner" (TPA stands for Third Party Access). This goal refers to a design option in which the PPC Distribution could offer access to its distribution network to third parties, i.e. independent power producers. Part of the result of the application of the elicitation process to this goal is shown in Figure 1.

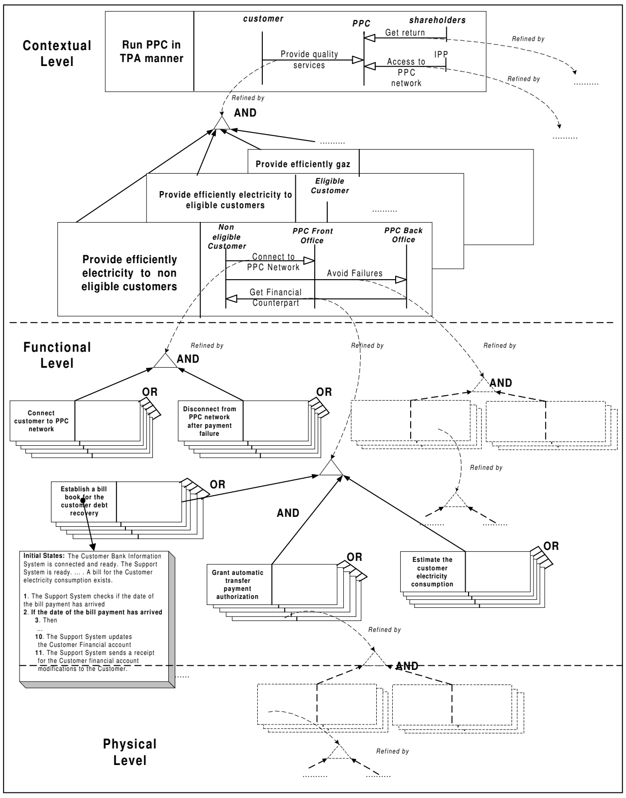

Figure 1. An excerpt of the Requirement Chunks of the ELEKTRA case study

In contrast, in the absence of the application of the approach, first the PPC distribution processes were described. From these, the operational goals were identified which were then clustered together to yield the final hierarchy, a part of which is shown in Figure 2. The total hierarchy consists of about 200 goals.

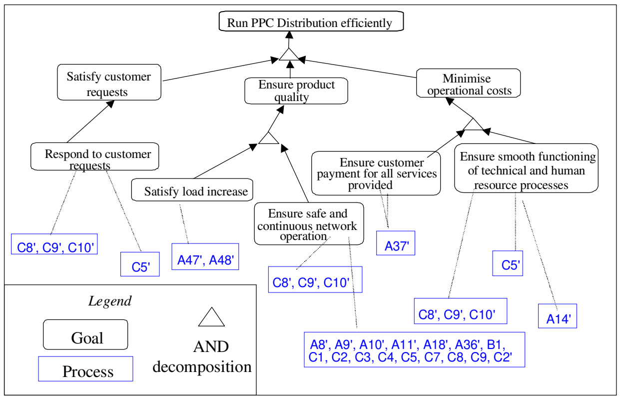

Figure 2. From processes to goals : elements of the final goal hierarchy

Issue 1 : Initial Goal Identification

In the application of the CREWS approach, the difficulty of identifying initial goals was low since the approach requires only a small collection of goals which can be treated as a single higher level goal. In contrast, the determination of operational goals and their clustering to yield the final 200 goals was not a straight-forward task and incorrect clustering had to be determined and removed. In fact, three different ways of clustering were tried out before the goals could be identified (see [9] for details). Also, considerable attention had to be given to remove redundant goals and to identify missing goals. All this was time-consuming and diverted attention from the main task of goal identification.

3.2. Goal analysis

3.2.1 Goal formulation. Goal analysis is concerned with the formulation of goal alternatives and subsequent reasoning to select the right alternatives. In order to do this, the CREWS approach expects that the informal goal statement will be brought into a form that is conducive to performing goal analysis. This form is as follows :

Verb <object> <result> <source> <destination> <means> <manner> <referent> <beneficiary> <time> <location> <quality> <quantity>

The use of this template helps in reformulating the preliminary informal goal statement into a more accurate definition. By focussing attention on possible values of the different parameters and their compatibility, the template helps in the generation of alternative goals.

Let us take the ELEKTRA preliminary goal statements:

(a) Customer Servicing
(b) Provide Electricity
(c) Disconnect Customer

For (a), the verb of the template is "Serve". Thereafter, reasoning on the parameter <beneficiary> of the template leads to the introduction of two types of customers, namely, eligible and non-eligible as follows :

Serve $_{\text{verb}}$ (eligible customers) $_{\text{beneficiary}}$
Serve $_{\text{verb}}$ (non- eligible customers) $_{\text{beneficiary}}$

Further analysis of the beneficiary shows that there can be eligible customers from other European countries. This lead naturally to the definition of a goal to attract such customers, i.e., to :

Attract $_{verb}$ (European eligible customers) $_{beneficiary}$

For (b), reasoning on the <means> leads to the following goals :

Provide $_{verb}$ (electricity) $_{object}$ (to our customers) $_{destination}$ (using PPC network) $_{means}$

Sell $_{verb}$ (access) $_{object}$ (to other parties) $_{destination}$ (using PPC network) $_{means}$

For (c), similar reasoning on the <time> leads to :

Disconnect $_{verb}$ (customer) $_{beneficiary}$ (upon customer request) $_{time}$

Disconnect $_{verb}$ (customer) $_{beneficiary}$ (upon PPC decision) $_{time}$

Thus, it can be seen that the template helps in a more accurate definition of a goal and, in addition, focussing on the parameters helps in finding other relevant goals.

Issue 2 : Goal Formulation

The initial goal statement is usually rather imprecise and sketchy and can be interpreted in many ways. The exact meaning of the goal gets clearer and clearer as the elicitation process proceeds through scenario authoring, goal refinement and goal decomposition. However, our experience is that it is best to make a precise, formal statement of the goal as early as possible in the RE process. Furthermore, all the parameters of the goal template are not necessarily needed in every goal. Our approach was improved to classify goal templates on the linguistic property of the verb [21] and it so turns out that all members of a class have the same set of parameters. This provides the potential for introducing guidance in goal formulation, a property that has been exploited in the CREWS L'Ecritoire prototype.

3.2.2 Exploration of design alternatives. In order to systematise goal finding, the CREWS approach suggests that the values of the different parameters should be determined independently of each other. It is only once all values of all parameters have been determined that the dependency between these values should be considered. This facilitates the generation of alternatives from which the appropriate ones can be selected.

Thus, starting from the goal : "Get $_{verb}$ (financial counterpart)$_{result}$ (from PPC customers) $_{source}$ (based on meter readings) $_{referent}$", we enumerate all possible means (being currently used or that could be used in the future), all possible referents, sources and time cycles. We obtain the following table (future possibilities are in *italics*).

Means	Referent	Time	*Source*
Private agencies PPC offices Post offices Bank transfer *Pre-paid cards* *WEB based interfaces*	Based on meter readings *Based on annual consumption* *Based on package price*	Every two months Every month *Every year*	*Non eligible customers* *Eligible customers*

Combining all these values leads to the construction of 108 different goals. Examples of such goals are given in the list below.

1. Get $_{verb}$ (financial counterpart) $_{result}$ (every month) $_{time}$ (from eligible customers) $_{source}$ (based on annual consumption) $_{referent}$. (by bank transfer) $_{means}$
2. Get $_{verb}$ (financial counterpart) $_{result}$ (from non eligible customers) $_{source}$. (using pre-paid cards) $_{means}$

In order to remove meaningless goals, it is suggested that a pair wise combination between parameters should be performed to exclude meaningless or contradictory ones. In the previous example, when combining the means and time parameters, if the payment is made using "pre-paid cards", the time is meaningless. Similarly, when combining the parameters, means and referent, if the payment is made using pre-paid cards, the referent is meaningless. This leads to a reduction of the number of possibilities from 108 down to 92. A complete analysis leads to reduce the number of possibilities to around 40.

Notice that the use of the template for finding alternative goals is applicable at any abstraction level. At the contextual level, the use of the template leads to elicit design options. For example, the use of the template on "Provide quality services" (see Figure 3.1) helps to investigate different types of design options such as. "Provide efficiently electricity to non-eligible customers", "Provide efficiently electricity to eligible customers", "Provide efficiently gas to non-eligible customers", etc. Each design option identifies a number of services such as "Connect to PPC network", "Avoid failures" and "Get financial counterpart" for the design option "Provide efficiently electricity to non eligible customers" (Figure 1). At the functional level, the alternatives describe the different possible realisations of a service. The different ways by which payments of electricity consumption can be performed, as identified above, are examples of functional alternatives. At the physical level, the alternatives refer to the different ways in which a function can be implemented. For example, the WEB payment alternative can be implemented using either the Internet or through an Intranet.

Issue 3 : Exploration of Alternative Designs

Our experience is that focussing the attention of engineers on key factors and then providing automated support facilitates the envisionment of a large number of alternative designs. In the BPR domain, this is crucial for the envisionment of the future system. However, this exhaustive generation of alternatives is very difficult to practice manually. In the example discussed above, the manual approach came up with only 7 alternatives whereas 108 alternatives were generated using the CREWS approach, out of which eventually only 40 were relevant. Evidently, there is a combinatorial explosion problem in the generation of alternatives. Yet, from the RE point of view, the larger the number of alternatives explored, the better it is. Our experience is that it is not possible to focus attention of stakeholders on a very large number of alternatives, typically above 20. Therefore, it was necessary to modify the approach to mitigate the combinatorial explosion. Two solutions were incorporated, (a) automatic elimination of goals (those having contradictory parameters), and (b) goal classification to limit the number of goal parameters.

3.3. Scenario authoring

Once a goal has been selected, a scenario is authored for it. In this movement from goals to scenarios, the scenario associated with a goal reflects the dynamics of the system for achieving it. Therefore, the goal gets grounded in reality : if there is a dynamic for achieving it then the goal must be a realistic one. Additionally, the scenario specifies a way of operationalising the goal. In this way the goal-scenario coupling mitigates the problem of fuzziness of goals and goal operationalisation as reported in the literature [20, 1].

3.3.1 Finding Whether Goals are Realistic or Not. The writing of a scenario can lead to discover that the associated goal is not realistic. For instance, let us consider the ELEKTRA goal "Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{result}$ (with a meter installation)$_{manner}$ (within 2 working days) $_{time}$" and the corresponding scenario partly given on the top of figure 3.

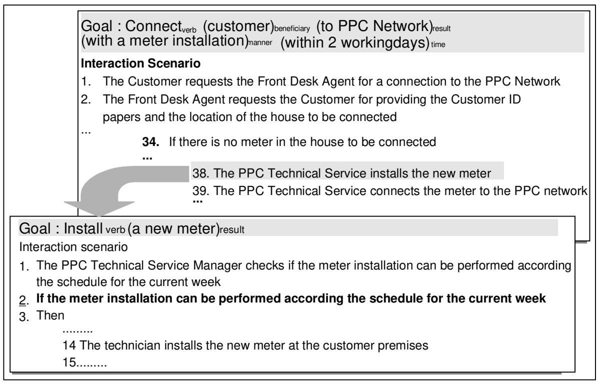

Figure 3. An example of an interaction refined by a goal and its associated scenario

In the application of the approach, non-operationalisable interactions in a scenario are considered as goals which have their own scenarios (see section 3.4). This is the case of interaction 38 in the above scenario. The goal derived from this interaction along with its associated scenario is given in the bottom of figure 3.

This scenario shows that the installation of new meters depends on the weekly schedule of the PPC Technical service (see condition number 2). If the schedule of the current week does not allow this then, the time constraint cannot be met which shows that the goal "Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{result}$ (with a meter installation)$_{manner}$ (within 2 working days) $_{time}$" is not realistic.

Issue 4 : Finding the Right Goal

The mere formulation of the goal after goal analysis does not necessarily lead to an acceptable goal. Experience shows that scenario authoring is a way to make the goal more concrete which, in turn, helps in making the goal formulation clearer and more accurate. This leads to finding the right goal.

Goal : Connect $_{verb}$ (customer)$_{beneficiary}$ (to PPC Network) $_{result}$

Interaction Scenario

1. The Customer requests the Front Desk Agent for a connection to the PPC Network
2. The Front Desk Agent requests the Customer for providing the Customer ID papers and the location of the house to be connected
3. The Customer provides to the Front Desk Agent the Customer ID papers and the location of the house to be connected

...

24. If there is a meter in the house to be connected

25. Then

...

33.If a remote connection is feasible

34. Then

35. The Front Desk Agent requests the Support System for performing the connection

...

Figure 4. Detailing the goal "Connect Customer to PPC network" with a scenario

3.3.2 Removing Goal Fuzziness. In our case study, the scenario shown in figure 4 was authored for the goal "Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{result}$".
The scenario shows one case of connection (see line 33 of Figure 3.4), that of using a remote connection for the customer. Thus, the scenario helps in identifying the parameter <manner> of the goal which leads to a more accurate formulation of the goal as follows :
Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{result}$ (through a remote connection) $_{manner}$

Issue 5 : Scenario Fitness

In order to properly resolve the issue of finding the right goal, the related issue of authoring a fit scenario must be raised and resolved. We conducted an empirical study to evaluate the effect of using guidelines to write scenarios on their fitness. Over one hundred and twenty people formed the target group of the study. From this study we found two important results (a) that guidelines for advising on the style and contents of the scenario significantly improve scenario fitness, and (b) that guidelines do not guarantee scenario fitness as a number of errors and ambiguities can still be found in scenarios. Thus, in addition to providing guidelines, it seems essential to develop suitable techniques to check for scenario fitness, for example by ensuring scenario completeness and removing ambiguities.

3.4 Goal elicitation through scenario analysis

Once a scenario has been authored, it is analysed to yield goals. Analysis is based on three different strategies, refinement, composition, and alternative discovery strategies. In this movement from scenarios to goals, the attempt is to discover realistic goals. Indeed, since a scenario describes a concrete and relevant way of realising a goal, any strategy that helps in discovering goals through scenario analysis is likely to produce the right goals.

3.4.1 Refinement Strategy. Consider the ELEKTRA goal "Provide $_{verb}$ (efficiently) $_{manner}$ (electricity) $_{object}$ (to customers) $_{beneficiary}$". The first statement of the associated scenario represents a service that the customer expects from PPC, namely, a connection to the network. This service cannot be directly operationalised. The refinement strategy suggests that such a service should yield a new goal at a lower level of abstraction than the original goal. Thus, as depicted in figure 5, a new goal, "Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{result}$" is defined and its scenario is authored.

Similarly, in this latter scenario, the statement 38 is an action which cannot be directly realised. Again a new goal, "Install $_{verb}$ (a new meter)$_{result}$", is defined and its scenario is authored. The refinement strategy is applied recursively till no new goals are discovered.

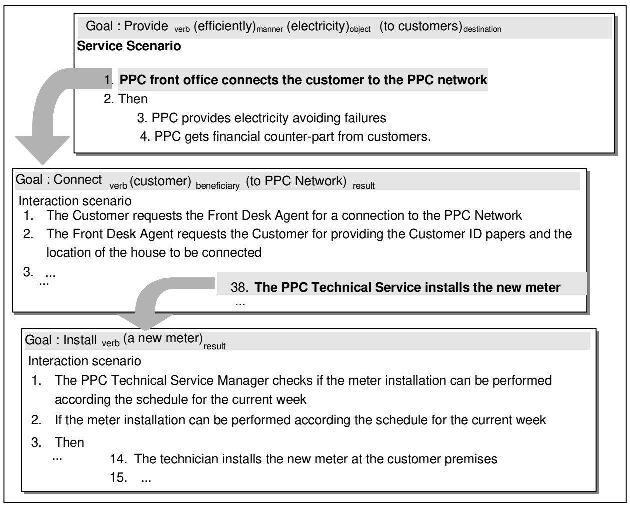

Figure 5. The successive refinements of a high level goal

Issue 6 : Change in level of abstraction

As pointed out in [28], goal operationalisation often requires a change in the level of abstraction. Our experience is that finding refined goals was difficult in some cases. This is similar to the experience of [5]. This difficulty was reduced by introducing in our approach, three pre-defined levels of abstraction, contextual, functional, and physical. These three levels provide a framework for the gradual transformation of high level enterprise goals into goals that could be related to business processes. The contextual level was found to be particularly useful in ELEKTRA because it provided a means to concretise high level goals by service scenarios that identify the key agents and the services that one requires from the other. In fact, a number of levels of abstraction within both the contextual and service levels were found necessary.

3.4.2 Composition Strategy. Consider the same ELEKTRA goal " Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{result}$ ". An application of the composition strategy to this goal results in three ANDed goals :

Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{result}$
Disconnect $_{verb}$ (customer) $_{beneficiary}$ (from PPC network) $_{source}$ (upon customer request) $_{time}$
Disconnect $_{verb}$ (customer) $_{beneficiary}$ (from PPC network) $_{source}$ (upon PPC decision) $_{time}$

Thus, the composition strategy helps in discovering the various functions of the system, i.e., the various use cases [14].

As another example, when we applied the composition strategy to the goal, "Get financial counterpart", we found the following ANDed goals :

1. Bill $_{verb}$ (customer) $_{beneficiary}$ (with respect to electricity consumption) $_{source}$
2. Get $_{verb}$ (financial counterpart) $_{result}$ (from customer) $_{source}$
3. Grant $_{verb}$ (automatic transfer payment authorization) $_{result}$
4. Revoke $_{verb}$ (automatic transfer payment authorization) $_{result}$
5. Estimate $_{verb}$ (the customer electricity consumption) $_{object}$
6. Establish $_{verb}$ (a bill book for customer debt recovery) $_{result}$

Issues 7 : Discovering Full System Functionality (Use cases)

Evidently, the discovery of full system functionality is of the essence. However, it seems to us that stakeholders have a tendency to concentrate on the most obvious functionality to the exclusion of others. It is therefore necessary to make them focus on discovering all functionality, complementary and supplementary. Whereas the composition strategy helps in the former, the latter is left unaddressed in our approach. We consider this, however, to be an important issue to be resolved. An associated problem of discovering system functionality is that of tracking. This arises when the discovery of a functionality is not accompanied by establishing its relationship with other functionalities. Our experience in this matter is similar to that of [5] in that it is difficult to relate goals spread over large parts of the goal hierarchy to system functionalities. In our approach, the use of the composition strategy naturally generates ANDed goals which by definition are goals identifying functionalities.

3.4.3 Alternative strategy. Again, the ELEKTRA goal, "Connect $_{verb}$ (customer) $_{beneficiary}$ (to PPC Network) $_{object}$ ", can be operationalised in seven different manners. These seven alternatives are discovered by the use of the alternative strategy and are ORed to each other. The seven manners are given below :

1. (through a remote connection) $_{manner}$
2. (when a remote connection is not feasible) $_{manner}$
3. (with a meter installation) $_{manner}$
4. (with a meter characteristics modification) $_{manner}$
5. (with an installation creation made by PPC) $_{manner}$
6. (by sub-contracting the installation creation) $_{manner}$
7. (by treating the exception "the customer is written-off") $_{manner}$

In contrast, in the absence of the application of the alternative strategy, only 3 alternatives, namely, "Supply low voltage customer with electricity", "Reconnect meter", and "Modify installation" were discovered. Whereas it is possible to treat the second goal as equivalent to "Reconnect meter" ; the first, third, fifth and sixth goals as "Supply low voltage customer with electricity" ; the fourth as "Modify installation", the seventh goal has not been discovered at all.

Issues 8 : Finding Functionality (Use Case) Variants

It has been recognised [5] that the process of identifying variations in use cases is ad-hoc and unsatisfactory. As seen in the example above, such processes are unable to discover all possible variants of a use case. On the other hand a systematic use of the alternative strategy does discover more variants than ad-hoc processes. It is of course very difficult to say whether all possible variants have been discovered or not. Our example also shows that the granularity of the variants discovered by the ad-hoc process is not uniform and is also coarser than that of the systematic process. The former calls for a detailed examination of the variant itself whereas the latter leads to re-engineering of the goal hierarchy, a time consuming activity.

3.4.4.Global view on goal discovery and operationalisation. Aside from the specific issues concerning refinement, composition, and alternative, it is necessary to look at the global issues involved in the goal operationalisation and elicitation process :

Issue 9 : Mastering Goal Operationalisation

We found that the separation of refinement, composition, and alternative strategies provides a way of mastering the complexity of the goal operationalisation and elicitation process. Refinement looks for goals which allow the operationalisation of a higher level goal. The alternative strategy helps in discovering different manners for achieving the same goal which can be seen as the different variations of a use case. The composition strategy helps in discovering goals which are necessary for the entire system to function such that each goal corresponds to a different use case. Thus the attention of the stakeholders is focussed on one problem at a time.

Issue 10 : Systematised Goal Discovery

We found that scenario analysis is a powerful technique of eliciting goals. In our case study, starting with the scenario associated to “Run PPC in TPA manner” (Figure 1) and the three actions, namely, “Provide quality services”, “Get return”, and “Access to PPC network”, 80% of all goals were discovered by successive goal elicitation through scenario analysis. In fact, these are a subset of the total goals generated by the application of the three discovery strategies and were selected in the final solution because they were considered to be the right ones.

4. Discussion and concluding remarks

The table below summarises the issues discussed in section 3 and identifies their source, i.e. whether they arise from goal-driven, scenario-driven, or goal-scenario coupling techniques. It can be seen that in trying to resolve the various issues, the coupling-driven approach raises a fresh issue, that of scenario fitness.

We feel that the relative importance of these issues can vary from problem to problem. In the BPR case study the two most important issues were exploration of alternative designs and systematised goal discovery. The former was crucial because it is best to consider as large a set of alternatives as possible when envisioning the future system. The importance of the latter was due to the relatively large size of the case study.

Issue No.	*Issue*	Goal-driven	Scenario -driven	Coupling-driven
1	Initial Goal Identification	X		
2.	Goal Formulation	X		
3.	Exploration of Alternative Designs	X	X	
4.	Scenario Fitness			X
5.	Finding the Right Goal	X		
6.	Change in Level of Abstraction	X	X	
7.	Discovering Full System Functionality		X	
8.	Finding Functionality Variants		X	
9.	Goal Operationalisation	X		
10.	Systematised Goal Discovery	X		

Even though the issues brought out in our case study have convergence with similar experience reports with either goal [28, 2] or scenario driven [5] approaches, our experience in the application of the coupling-driven approach in resolving these issues is worth highlighting. This experience is different on account of the bi-directional coupling between goals and scenarios found in the CREWS-L’Ecritoire approach.

This bi-directional coupling makes it possible for the elicitation process to be proactively and systematically guided through an iterative cycle consisting of goal-analysis, scenario authoring, goal elicitation through scenario analysis. Quite obviously, this leads to systematised goal discovery thereby helping to resolve the tenth issue in the table above. As a side effect of this, the first issue of initial goal identification is also addressed. This is because the systematisation of the process permits to start with one single high level goal and its associated scenario and to progressively and systematically derive the entire hierarchy of goals from this scenario. Thus we could avoid dealing with a large number of goals at the same time.

When traversing the bi-directional coupling in the forward direction, i.e. from goal to scenario, we found that the scenario attached to a goal helps in finding the right goal. Thus the fifth issue can be addressed. This is obtained in exchange of careful scenario authoring and analysis. Our experience, that a large majority of goals can be discovered from scenario analysis, shows that perhaps this expense is worth the pay-off.

In the reverse direction the scenario associated to a goal facilitates (a) mastering the change in levels of abstraction (issue 6), (b) discovering full system functionality (issue 7) and (c) finding functionality variants (issue 8). Effect (a) happens because the scenario, by describing the dynamics of a goal, subsumes its operationalisation. Therefore, it becomes possible to develop mechanisms such as our refinement strategy to support the change in level of abstraction and to help in discovering goals at a lower level of abstraction. Similarly, effects (b) and (c) occur on account of statements which permit reasoning on dependent and alternative dynamics respectively. Therefore strategies such as our composition and alternative ones can be defined to deal with the discovery of dependent and alternative dynamics.

Thus it can be seen that movements in the forward and reverse directions are mutually reinforcing. The meeting point of these movements is a scenario. Clearly the analysis of a scenario becomes crucial and therefore, the issue of scenario fitness arises in the coupling-driven approach. In fact, in our experience, almost as much work was required in developing strategies for scenario analysis as for ensuring scenario fitness.

In this paper we have not considered the use of automated support in the case study. This support was available in two forms. The first was a the CREWS L’Ecritoire software tool which provides guidance in carrying out the various steps of the elicitation process [27]. The second form of automated support was used in conducting meetings and brain-storming sessions. These sessions were supported by “GroupSystems®”, a CSCW tool. Automated support increases requirements engineer productivity and promotes method systematisation with its accompanying benefits.

5. References

[1] Anton A.I., McCraken W.M., Potts C., *Goal Decomposition and Scenario Analysis in Business Process Reengineering*, Advanced information System Engineering, 6th Int. Conf. CaiSE'94 Proc., Utrecht, The Netherlands, 6-10 June 1994, pp 94-104, 1994.

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

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

[4] J. M. Caroll, "*The Scenario Perspective on System Development*", in Scenario-Based Design: Envisioning Work and Technology in System Development, Ed J.M. Carroll, 1995.

[5] Cockburn, *Structuring use cases with goals*. Technical report. Human and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT.TR.95.1, http://members.aol.com/acocburn/papers/usecases.htm, 1995.

[6] Dano, H. Briand, F. Barbier, *A use case driven requirements engineering process*. Third IEEE International Symposium On Requirements Engineering RE'97, Antapolis, Maryland, IEEE Computer Society Press, 1997.

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

[8] 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.

[9] ELEKTRA consortium, *DEMETRA : System Design Specification for PPC*, ELEKTRA deliverable, March 1998.

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

[11] Hammer M. Champy J., "*Re-engineering the corporation : a manifesto for business revolution*", Harper Collins publishers, inc., New York, 1993.

[12] Peter Haumer, Klaus Pohl, Klaus Weidenhaupt, *Requirements Elicitation and Validation with Real World Scenes*, to appear in IEEE Transactions on Software Engineering, Vol. 24, No. 12, Special Issue on Scenario Management, December 1998.

[13] C. H. Holbrook, *A scenario - based methodology for conducting requirements elicitation*. ACM SIGSOFT, Software Engineering Notes Vol. 15, N° 1, pp. 95-104. January 1990.

[14] I. Jacobson, *The use case construct in object-oriented software Engineering.* In ''Scenario-based design: envisioning work and technology in system development'', John M. Carroll (ed.), John Wiley and Sons, 309-336, 1995.

[15] 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 Requirements Engineering RE'97, Antapolis, Maryland, IEEE Computer Society Press, pp. 44-53, 1997.

[16] P. Loucopoulos, *The F^3 (From Fuzzy to Formal) view on requirements engineering.* Ingénierie des Systèmes d'Information, Vol. 2, N° 6, pp. 639-655, 1994.

[17] 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.

[18] K. Pohl, P. Haumer, *Modelling contextual information about scenarios*. Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'97, Barcelona, pp.187-204, June 1997.

[19] C. Potts, K. Takahashi, A.I. Anton, *Inquiry-based requirements analysis*. In IEEE Software 11(2), pp. 21-32, 1994.

[20] 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'97 , Barcelona, pp. 15-28, June 1997.

[21] 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.

[22] C. Rolland, C. Ben Achour, *Guiding the construction of textual use case specifications*. Data & Knowledge Engineering Journal Vol. 25 N° 1, pp. 125-160, (ed. P. Chen, R.P. van de Riet) North Holland, Elsevier Science Publishers. March 1997.

[23] C. Rolland, C. Souveyet, C. Ben Achour. "*Guiding Goal Modelling using Scenarios*". To be published in IEEE Transactions on Software Engineering, Special Issue on Scenario Management, Vol. 24, No. 12, Dec. 1998**.**

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

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

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

[27] M. Tawbi, C. Souveyet, C. Rolland, *L'ECRITOIRE a tool to support a goal-scenario based approach to requirements engineering*, Submitted to Information and Software Technology journal, Editor : Martin Shepperd, Puplisshers : Elsevier Science B.V, 1998

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

[29] K. Weidenhaupt, K. Pohl, M. Jarke, P. Haumer*, Scenario usage in system development : a report on current practice*. IEEE Software, March 1998.

[30] E. Yu, J. Mylopoulos*, Using goals, rules and methods to support reasoning in business process reengineering*. Proceedings of the 27th Hawaii International Conference System Sciences, Maui, Hawaii, January 4-7, Vol. IV pp. 234-243, 1994.