Guiding the Requirements Engineering Process

Colette Rolland
Université de Paris Sorbonne
17, rue de la Sorbonne
75231 Paris Cedex 5
France

Naveen Prakash
Delhi Institute of Technology
Kashmere Gate, Old IG Block
110006 Delhi
India

Abstract

Despite a constant improvement of Information Systems methodologies and their partial automation in CASE environments, the level of guidance provided to developers is still low, especially in the early phases of information systems engineering. We argue in this paper that a more effective guidance to requirements engineers can be based on a process meta-model which allows to deal with a large variety of situations in a flexible, decision-oriented manner and supports different levels of granularity in decision making. The paper presents the process guidance meta-model and shows how it can guide the requirements engineer punctually -to take the right decision in the adequate situation- as well as globally, to control the decision flow.*

1: Introduction

The complexity of Information Systems (IS) needed by the user community implies a methodological support for their development. However, despite thirty years of methodologies [14, 15], tools practise and research, we know very little about the IS development process (see for example the recent in-depth studies of software development practices [15]) and how to guide it. On the other hand, the IS community and the Software Engineering community have a wide consensus that the quality of products depends on the process through these products are produced [1, 10, 20, 25]. This calls for an improvement of process meta-models and way-of-working definitions.

Ways-of-working are often not clearly defined at a too high level of granularity. Take as an example the high-granularity process model consisting of four steps to define an E/R schema :

- define entities
- when finished, define relationships
- when finished, assign attributes to entities
- when finished, attach further constraints

Even if they are clearly defined, ways of working are lacking flexibility and therefore are not used as intended. Consequently CASE tools are not able to provide an effective support during IS development. They mainly propose graphical interfaces to capture the IS product at different levels of detail and a repository support for storing, manipulating and documenting the products (specifications), but very little is offered to support the process of IS development itself.

The first idea of the approach presented in this paper is to base process guidance on a process guidance meta-model. We argue that the process guidance meta-model can deal with many different situations in a flexible, decision-oriented manner and can support different levels of granularity in decision making. The second contribution is to show how the process guidance meta-model can provide guidance punctually, at a given point in time of the IS development and also help in controlling the development flow, i.e. supporting the development process globally.

The approach focusses on the Requirements Engineering (RE) phase. It has being implemented as part of the NATURE requirements engineering environment [12] and being validated with the particular way-of-working of the F3 project [4].

In section 2, we present the basic process meta-model from which the process guidance meta-model presented in section 3 is derived. In section 4 we show how a specific way-of-working can be defined by instantiation of the process guidance meta-model. The specifics of local and global guidance is discussed in section 5. Finally we draw some conclusions in section 6.

2: The Basic Process Meta-Model

If we want a general frame to define and improve ways-of-working on a wide range of methodologies, we, first of all, need a process meta-model providing a generic set of concepts allowing a proper description of what occurs in a requirements engineering process, when it occurs, why and on what it happens. The basic process meta-model presented in this paper belongs to the class of process models referred to by Dowson as "decision-oriented models". However it emphasizes the notion of context in which a decision is taken, offers guidance in this context through possible actions but leaves freedom to the requirements engineer in the choice of actions and the selection of contexts as well.

* This research is supported in part by the EEC through the ESPRIT project F3 : From Fuzzy to Formal.

0-8186-6960-8/94 $04.00 © 1994 IEEE

Dowson [5] identifies three main classes of process models:

- activity-oriented models,
- product-oriented models,
- decision-oriented models.

Activity-oriented models come from an analogy with problem solving, i.e. finding and executing a plan of actions leading to the solution. They are sequential in nature and provide a frame for manual management of projects developed in a linear fashion. The first widely used model, the Waterfall model [26], falls into this category, along with the Spiral model [2] and the Fountain model [9] which try to eliminate the well recognized lack of flexibility of the Waterfall model.

The product-oriented process models represent the development process through the evolution of the product. They promote a view of development processes which is still centred around the notion of development activity but presents the advantage to link development activities to their output : the product. The ViewPoints model [7] belongs to this category as well as the development process model proposed in the European Software Factory (ESF) project [18].

A more recent class of process models follows a *decision-oriented paradigm*.The successive transformations of the product are looked upon as consequences of decisions. The process models of the DAIDA project [11] and of [19] fall into this category. Such models are semantically more powerful than the previous ones because they explain not only how the process proceeds but also why transformations happen.

Considering the limitations of the first two classes of process models [21], we chose to base our process modeling approach on the decision-oriented one. However, even the decision-based approaches offer only limited hints about when and how to decide on what. We recognize that requirements engineers react *contextually* according to the domain knowledge they acquire and react by analogy with previous situations they have been involved in. In order to take into account this very nature of the RE we have chosen to emphasize the contextual aspect of decisions by strongly coupling the context of a decision to the decision itself. The NATURE approach aims at capturing not only activities performed during the RE process but also why these activities are performed (the decisions) and when (the decision contexts).

Figure 1 illustrates the basic RE process model introducing its key-concepts and their relationships (with a binary E/R based notation). This contextual approach is an extension of the one presented in [8] and implemented in the ALECSI prototype [22].

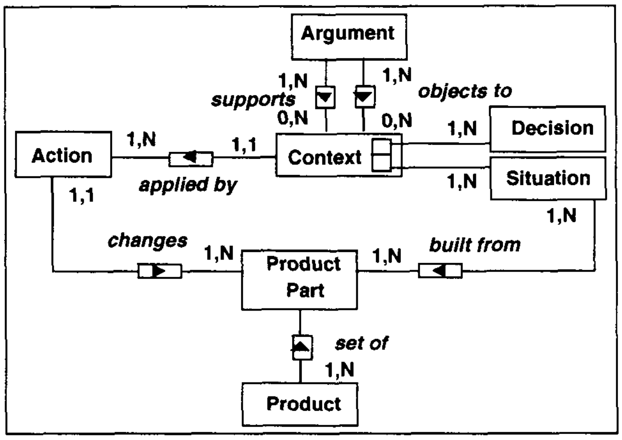

Figure 1 : The basic Process Model

A *situation* is most often a part of the product it makes sense to make a decision on. Situations are thus built from parts of the product undergoing the RE process. At the beginning of the process, the product is made of the collection of the users' requirements ; At the end of the process, the product is a specification of the IS that satisfies some quality criteria. Situations in RE can be of various granularity levels. They can be either very atomic like a cardinality on a role in an E/R schema ; Or they can be coarse-grained like the whole specification under development.

Moreover, situations can also be built from existing, reusable parts of previously developed products. [23] presents a model of the RE product and details the building of situations from its parts.

A *decision* reflects a choice that a requirements engineer makes at a given point in time of the RE process. A decision can be looked upon as the intention of product transformations.

A *context* is the association of a situation and a decision made on it.

An *action* performs a transformation on the IS description. It is the materialization of a decision. Performing an action changes the product and may generate new situations, subjects to new decisions.

An *argument* allows to motivate the making of a decision on a given situation or to object to this decision making.

The five concepts are exemplified in figure 2 where we consider part of the RE product for the Air Traffic Control case study provided by one of the users companies of the ESPRIT project F3.

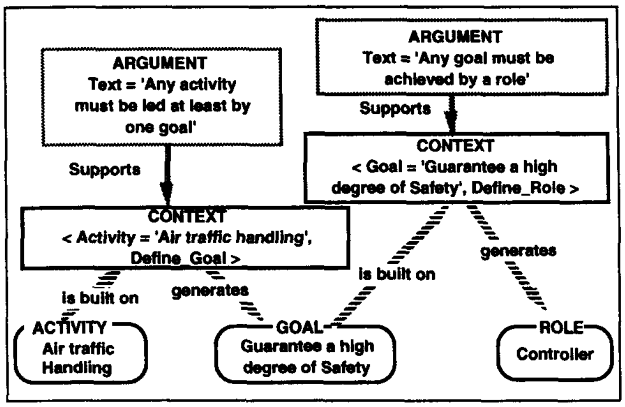

Figure 2 : example of contexts in the ATC case study

The F3 RE product is composed of 4 sub-models, namely the Enterprise Model (EM), the Non Functional Requirements Model (NFRM), the Human Computer Interface Model (HCIM) and the Information System Model (ISM). The first context considered in figure 2 is composed of the activity-typed EM element called 'Air Traffic Handling' and the decision 'Define-Goal'. This decision is supported by the argument stating that 'Every activity must be led at least by one goal' and triggers the action of creating the new goal 'Guarantee of a High Degree of Safety' in the Enterprise Model. The change induced by this action in the RE product raises a new situation, namely the goal 'Guarantee of High Degree of Safety' on which the decision to 'Define Role' is supported by the argument 'Every goal must be achieved by a role'. Applying the decision results again in a change of the EM by adding a new element : the role 'Controller'.

3: The Process Guidance Meta-Model

The basic process meta-model defines the building block of the RE process, namely the quadruplet <situation, decision, argument, action> relating a situation to one of the possible decisions to be taken in this situation, the arguments which object or support this decision and the corresponding action. However it lacks features to deal with aspects such as: which decision should or could be selected among the set of possible ones in the current situation, which context should be considered next, is there any prescription to be followed in this particular situation or to control decision flows? It lacks guidance and prescriptive features which are usually intended to be provided by ways-of-working of given methodologies.

The process guidance meta-model is a specialization of the basic process meta-model refining the basic process meta-model concepts and including new ones to provide the means of an efficient guidance in the decision making process of developing a RE product.

Drawbacks arising from the lack of the basic process meta-model for process guidance are essentially of three types. Firstly, all decisions are equally treated, although some of them are more complex that others. Secondly, the process modeling concepts of the basic model do not allow the representation of heuristics that can greatly help the Requirements Engineer in his decision-making process. Finally, the basic process meta-model is not capable of representing different approaches in the RE process such as the top-down or bottom-up ones. These approaches have impacts on the ordering of decisions made throughout the RE process.

Therefore the extensions we propose are themselves of three kinds. We will successively introduce different granularity levels through the specialization of the "context" concept to handle the complexity and heuristics representation needs. Then, we will introduce the process chunk as a meaningful process unit, i.e. the "process element". Finally the explicit modeling of process elements flows will allow the representation of approaches in the RE process.

The particular process modeling requirements that guidance raises can be reflected by extending the basic process meta-model of section 2, as illustrated by the figure 3 in which the white boxes correspond to the basic process model concepts.

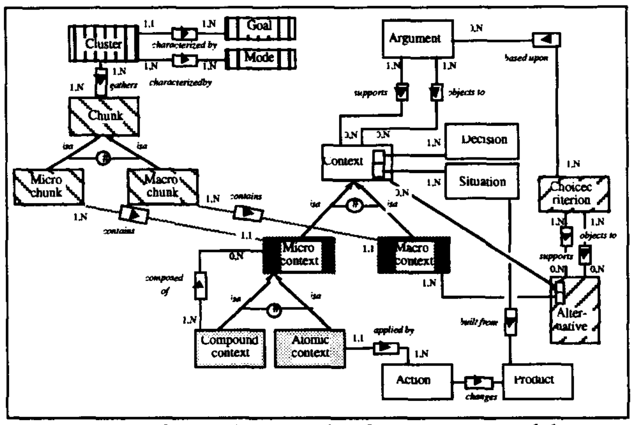

Figure 3: The Process Guidance Meta-Model

3.1: Handling granularity in decision-making: context specialization

First, contexts are subtyped into micro and macro contexts and micro-contexts are themselves subtyped into compound and atomic contexts.

According to the basic process meta-model, all the contexts are treated the same, they all are applicable by the performance of actions modifying the product. This view is a simplification of the kind of contexts that are encountered in reality. In fact, it is true that some contexts are directly applicable by actions, like the definition of a cardinality constraint or the creation of a new entity-type in the development of an E/R schema. But other contexts are more complex, like the partitioning of an entity-type into several, mutually exclusive subtypes. This former decision is decomposed as follows :

(1) subtype the entity-type

(1.1) create a new entity-type

(1.2) create the isA link between the initial entity-type and its new subtype

(n) repeat (1) as many times as there are subtypings to be done

(2) constrain the isA links by a disjunction constraint

(2.1) create the disjunction constraint

(2.2) link the disjunction constraint to the isA links

To handle this kind of granularity in decisions, we refine the *context* concept of the basic process model by adding the *atomic-context* and *compound-context* concepts, as illustrated in the figure 3. A compound-context is a context composed of contexts; contexts that cannot be decomposed are atomic-contexts. We represent the fact that component contexts can themselves be compound, leading to composition-based hierarchies of contexts, by adding a *composed of* relationship between *compound-context* and *context*..

Figure 4 gives the example, in the Air Traffic Control (ATC) case study, of the decomposition of the compound micro-context <[Air-Traffic-Handling],[Hinders],[When Aircraft is in Approach Response Time is Critical]; Partition] into seven atomic contexts including each, a sub-micro situation of the initial one and the corresponding micro-decision.

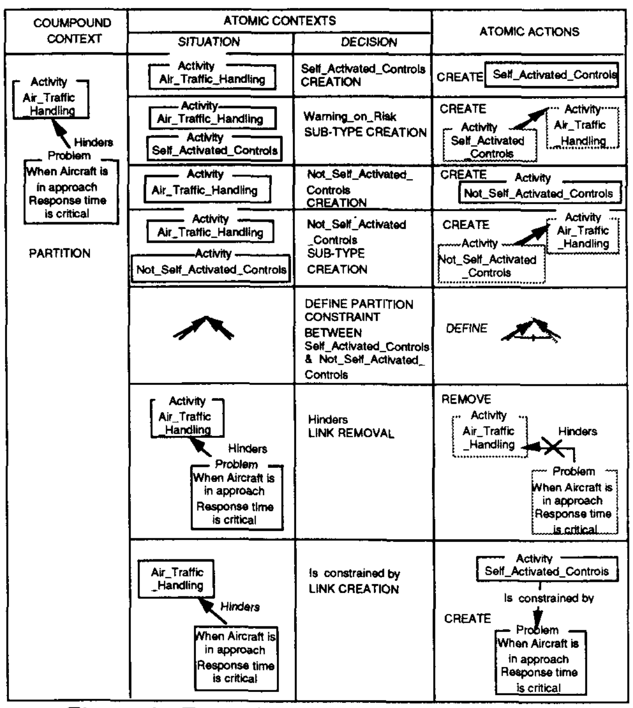

Figure 4 : Example of compound micro-context decomposition

Each micro-decision is implemented through the execution of one action which creates or removes elements of the EM under construction. For example the first atomic context is composed of the situation 'Air-Traffic-Handling' and the decision 'Self-Activated-Controls Creation' which results in the creation of the sub-type 'Self-Activated-Controls' of the Activity 'Air-Traffic-Handling'.

The second refinement of contexts we introduce is the creation of two new subtypes : *micro-context* and *macro-context*, both subtypes of the *context* concept. This extension of the basic process meta-model intends to represent the fact that some contexts can be applied by a unique set of product transformations while other contexts have several alternative ways to be applied. Contexts of the first kind are called micro-contexts (atomic- and compound-contexts as introduced above are of this kind) . Contexts of the second kind require that the Requirements Engineer chooses among a set of candidate alternatives, they are called macro-contexts.

For example, the context whose decision is to add an entity-type in a schema may have as alternatives the one that creates the new entity-type from scratch and another one that reuses an entity-type already created in the course of a previous RE process. In turn, this latter "reuse" context may have as alternatives the one to perform a "domain-based reuse" [16] and the one to perform a "similarity-based reuse" [24].

Alternatives of a macro-context are contexts too, thus contexts may share an alternative-relationship between them, leading to alternative-based hierarchies of contexts.

As illustrated in figure 3, along with the specialization of contexts, this extension to the process meta-model is ensured by the *alternative* relationship between *macro-context* and *context*. Choice criteria are associated to alternatives to help the decision-making process by supporting or objecting to the different alternatives of a given macro-context. These choice criteria are combinations of the arguments attached to the contexts as presented in section 2. A choice criterion aims at providing priority rules to select one alternative among the ones of a macro-decision.

Considering again the ATC case study we can see in figure 5 that the 'Partition' decision with its associated situation (detailed in figure 4) is one of the alternatives to 'Improve' the Activity 'Air-traffic-Handling' but is, itself, one of the possible ways to 'Improve' the 'Current-EM-Product' under development. If we remember that the F3 RE product has a number of components we finally end-up with the first level of the hierarchy depicted in figure 5, showing that to 'Improve' the 'Current-Product" under development, it is possible for instance to 'Improve' the 'Current-IS-Sub-product' or the 'Current-EM-Sub-product' or to 'Improve-Inter-Model-Links' in the 'Current-Product'.

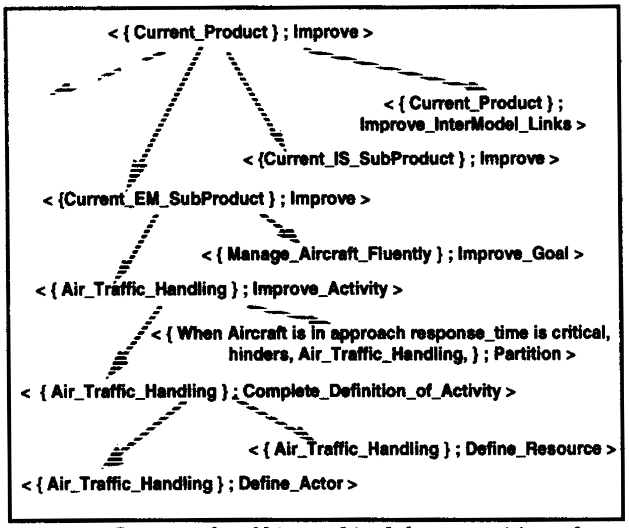

Figure 5: example of hierarchical decomposition of a macro-context

The introduction in the process meta-model of alternatives and choice criteria advises the requirement engineer on what can possibly be done in the current situation. The notion of alternative-based guidance leaves freedom to the designer. It even may suggest to take an alternative which is not in the set of pre-defined ones. We will see later on (section 6) that this is the way through which the process model- i.e. the intended way-of-working - can be improved throughout experiences. The meta-model also allows the representation of heuristics that help the Requirements Engineer by providing him with several motivated ways of applying decisions.

3.2: The chunk as the elementary RE process element

With the process meta-model extensions we just presented, a specific guided way-of-working can be defined as collections of micro-contexts (atomic or compound ones) and macro-contexts.

According to [6], atomic-contexts correspond to "process steps" (they have no visible substructure) while the composition- and alternative-hierarchies introduced above correspond to "process elements" : they are built from contexts like "elements" are built from "steps". With these composition- and alternative-hierarchies, the decision-making process is a process of hierarchical decomposition of compound- and macro-contexts. A compound-context is applied through the successive application of all its component contexts, until atomic-contexts are applied. A macro-context is applied through the recursive choice of its alternatives.

We call *chunks* these hierarchies. A *micro-chunk* is the composition hierarchy of a top-level compound-context. A *macro-chunk* is the embedding of the hierarchy of alternatives of a top-level macro-context and of the choice criteria associated to each level of the hierarchy. With this introduction of the *chunk* concept as the new unit of process definition, guided ways-of-working will be defined as *collections of chunks* and the process guidance will be based on a *library of micro and macro-chunks.*

Macro and micro chunks provide different types of guidance. Indeed there are two major differences between the macro-chunk and the micro-chunk : the first one lies in the absence of any alternatives in the latter and the second one is that a macro-chunk has no direct consequence on the product under development.

The macro-chunk corresponds to a situation which requires the exploration of alternatives in decision making. Each alternative is an approach or a strategy for the resolution of the issue being faced by the engineer in the current situation. The chunk supports the requirement engineer in the exploration of these strategies and in the selection of the most appropriate one. By definition, a macro-chunk offers a choice among a set of strategies all of which achieve the same purpose. In this sense, one can look upon the macro-chunk as being goal oriented.

A micro-chunk refers to a situation for which the strategy to be followed has been decided. It supports the implementation of the tactic associated with the strategy through the ordering of actions to be performed on the product under development. A micro-chunk is associated with a micro context, i.e., to a pair <situation, decision> which implies only one way of solving the issue it raises. One can say that the micro-chunk is only concerned with the manner in which the decision shall be implemented and not with exploring alternatives. The implementation of the decision of the micro-context will be performed by actions that transform the product under development.

The guidance provided by a micro-chunk is prescriptive. The chunk tells to the requirements engineer which actions to perform, and in which order in the situation at hand. Vice-versa the macro-chunk advises the engineer in the selection of alternatives existing in the situation he is faced to. It leaves him freedom to choose the appropriate alternative which can be a not yet recorded one.

3.3: modeling process elements flows

Contexts and chunks are dealing with punctual situations in the RE process. They support the local reasoning related to the treatment of a given context whatever it is, atomic, compound or macro. There is another kind of process knowledge which is required to control the flow of decisions as the RE process proceeds. This is achieved in the guidance meta-model by modeling the way chunks i.e. process elements can be meaningfully grouped and ordered.

We believe it is relevant to group chunks that together allow the application of a common intention or *goal.* We thus introduce the *cluster* concept into the process meta-model.

For example, let's consider the goal to complete a schema in an object-oriented method providing such concepts as "Class", "Event" and "Actor" (e.g. the O* method [3]). This goal is applicable through the specification of all the classes , all the events and all the actors of the schema, i.e. through the application of the corresponding three macro-chunks. These chunks are thus gathered into a "Complete_Schema" cluster.

Furthermore, we propose a second criterion to gathering chunks, which is the notion of *mode*.. We have identified two pairs of modes : *depth-first/width-first* and *top-down/bottom-up*.. Keeping the same example of an object-oriented methodology, it is possible to either prescribe that all classes must be specified before the specification of the events and actors can be undertaken, or it may prescribe that classes, actors and events specification can be undertaken in parallel. In other words, the process can proceed either in a depth-first manner or in a width-first one.

Similarly, it is possible to either prescribe that classes must be specified by creating them before defining their attributes or it may prescribe that attributes must first be defined and thereafter grouped to create classes. Here, classes can be specified either in a top-down manner or in a bottom-up one.

We model the influence of modes in the process by characterising a cluster not only by its goal but also by the particular mode it promotes. For a given goal, applicable by a same set of gathered chunks, there may be several clusters for each possible mode. For instance there can exist two clusters associated to the goal "Construct class": the one promoting a top-down fashion and the other guiding the process in a bottom-up manner. Multiplicity of clusters with the same goal allows to model variants in the way of engineering requirements. It introduces flexibility in the definition of ways-of-working.

Introducing clusters in the guidance meta-model allows to prescribe the way flows of contexts can be handled. The prescription is goal oriented and flexibility is brought by allowing the same goal to be achieved in different ways, modelled as different clusters. We will see in section 5 that additional freedom is gained by leaving the requirements engineer choose the context he wants to work on, at any moment in the RE process. We believe that such freedom is important because we do not know the precise order of activities. The process model must be modifiable during the actual process.

4: Intended Way-of-working Definition

According to our approach, process guidance will be based on a particular instantiation of the guidance process meta-model. This instanciation corresponds to the definition of the guided process model corresponding to the usual way-of-working provided by a given methodology. We will refer to it as the intended way-of-working. Such an instantiation can be sketched as follows.

First, contexts relevant for the various product models supported by the methodology have to be identified and subtyped into micro-contexts and macro-contexts. Macro-contexts are associated to their alternatives, supported or objected to by choice criteria (combinations of arguments). Micro-contexts have to be associated with either atomic actions or ordered sets of actions.

Then, contexts are respectively grouped into micro- and macro-chunks. This is simply made by encapsulating in a micro-chunk either the atomic action or the decomposition of the initial micro-context into atomic contexts. A macro-chunk embeds the decomposition tree leading to the refinement of an initial macro-context into micro-contexts through several decomposition levels .

We introduce a special context called "backtrack" which allows the selection of a previous context. This has the effect of "resetting" the process to a previous step.

Finally, clusters gather chunks and are characterized by their goal and their modes. Starting with the identification of a hierarchy of goals, the method engineer defines grouping of chunks to support a given goal with appropriate modes. As we mentioned before several clusters can be defined with the same goal to support various ways of proceeding.

The introduction of the chunk and cluster concepts allows, in particular, the definition of ways-of-working that embed RE heuristics. To be able to express heuristics is an important improvement in way-of-working definitions since it allows to capitalize experiences and permit that requirement engineers share their practical knowledge and methods.

We also believe that modeling the RE process according to our approach is not limited to its technical aspects, i.e. the evolution of the product, but can also tackle project-management issues. For example, macro-chunks could be defined to handle a situation in which a project deliverable completion exceeds its planned deadline. Then two possible alternatives could be suggested by the macro-chunk: to hire a highly-skilled manager to control the RE team or to double the team size. Corresponding choice criteria would then be associated to these alternatives.

5: Process Guidance

The process guidance meta-model suggests a RE process guided by the use of a library of *chunks* and *clusters.* Both are *context-based*.. Chunks provide punctual support to *handle contexts* which are more or less complex. Clusters help to *control the context flow.* This leads to the RE process has being an incremental process allowing to build the RE product progressively through the adequate ordering of context handling steps.

5.1: The spiral view

We thus, propose a global view of the RE process depicted in figure 6 as a spiral. Each turn of the spiral corresponds to one context handling activity guided by the appropriate chunk. It results in some product changes. The progression from one turn to another i.e. the angular movement in the spiral, represents the incremental production of the RE product. The radial movement, i.e. the unfolding, represents the extent of completeness achieved by the RE product. The flow represented by the progression from one turn to another will be guided by a cluster.

In the ATC example, as shown in Figure 6, turn1 is concerned with the decision to 'Improve-Activity' applied on the situation ' Traffic-Handling-Activity' which is an activity-typed element of the Enterprise Model built as a part of the F3 product. As we will see later on, performing the decision will result in the creation of a new component 'Guarantee a High degree of Safety' in the EM and the emergence of a new context aiming at refining this goal, that will be tackled in turn2. The macro-chunk <[Traffic-Handling -Activity]; Improve-Activity> will guide the requirements engineer in turn1. The cluster associated to the goal "Interpret EM element in terms of IS functions" will suggest to consider the context <[Guarantee a High Degree of Safety], Refine-Goal> in the next turn of the spiral.

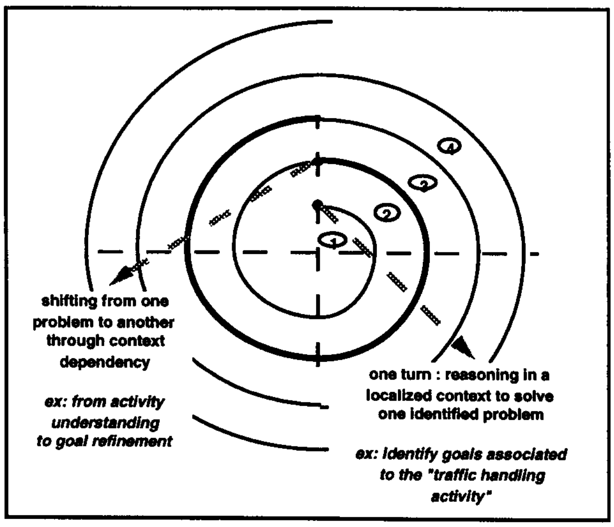

Figure 6 : A spiral view of the R.E. process

5.2: Structuration of a spiral turn

The input of each turn in the spiral is a context. The activity performed in the turn aims at refining the decision associated to the input context till it can be implemented through atomic actions of product transformation. In other words, the turn is guided by one process chunk. If the initial context is a macro-context the chunk provides the potential hierarchical decomposition of the input macro-decision. Therefore the process performed in one turn consists of selecting the adequate branch in the tree i.e. choosing at each level of the tree, the adequate alternative till a micro-context is selected. Then the micro-context is executed. If it is an atomic one this will result in only one action; if the micro- context is a compound one it implies the execution of a sequence of atomic actions leading to product transformations and raising new situations which are themselves subjects to decisions. We propose that the spiral turn includes the selection of the next context to be worked on. This results in the structuration of the spiral turn into *four quadrants* as shown in figure 7. Each quadrant corresponds to a phase in the guided process.

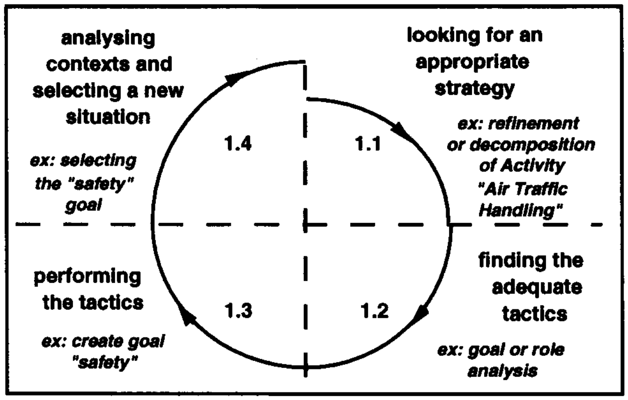

Figure 7: Decomposition of a spiral turn

The *first quadrant* aims at selecting the adequate strategy for solving the problem raised by the input context of the spiral turn. This is achieved by choosing among the alternatives proposed by the chunk associated to the input context. As an illustration Figure 7a shows the decomposition of the context <[Air-Traffic-Handling]; Improve_Activity > into a number of macro-contexts among which are <[Air Traffic Handling]; Decompose_Activity> and <[Air Traffic Handling]; Refine_Activity>. This means that there are at least, two ways of improving our understanding of the Air -Traffic-Activity : one is to decompose the activity into sub-activities and the other one is to refine it through goal analysis. Let us assume that the requirements engineer chooses the second of these alternatives.

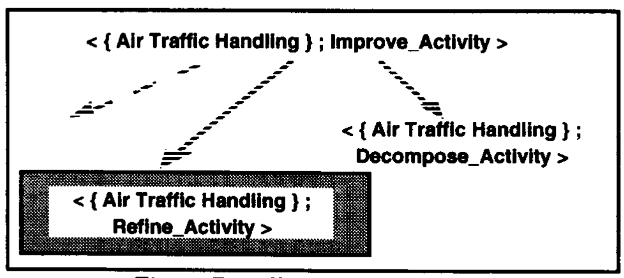

Figure 7a : Illustration of phase 1

In *quadrant 2* the macro-context resulting from phase1 will be decomposed further on in order to reach a micro-context. This will consist of selecting a branch in the hierarchical decomposition of the contexts encapsulated in the chunk associated to the spiral turn. This is illustrated with the ATC case study in Figure 7b. There are a number of possible alternatives to handle the 'Refine_Activity' decision of the context <[Air-Traffic-Handling]; Refine_Activity>; Choosing the <[Air-Traffic-Handling]; Define_Goal_of_Activity> requires further refinement. Among the two suggested decompositions of this context we choose the <[Air-Traffic- Handling]; Define_New_Goal_of_Activity> which is a micro-context that will be implemented in phase3.

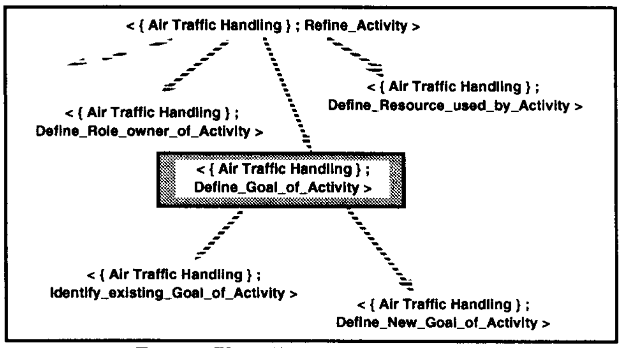

Figure 7b : Illustration of phase 2

Quadrant 3 is tactical. It is concerned with the implementation of the decision of the micro-context resulting of the decomposition process performed in quadrant 2. This is exemplified for the ATC case study in Figure 7c. The 'Define_New_Goal_of_Activity' decision requires four actions to be performed : the creation of the new component in the Enterprise Model, the goal ' Guarantee a High Degree of Safety' and the creation of its relationships with the other components of the model.

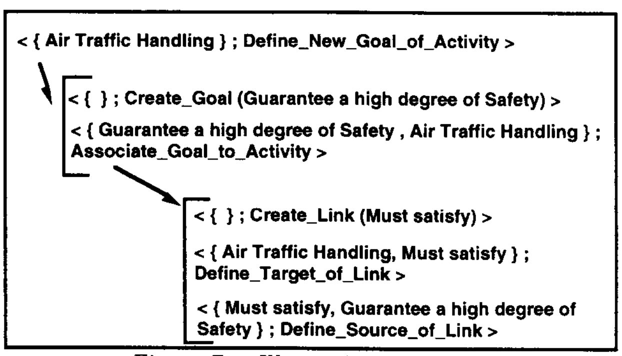

Figure 7c : Illustration of phase 3

Phase 4 has the objective to select the new working context . Performing the actions resulting of phase3 raises new contexts among which one may be selected. In the example in hand the creation of the component 'Guarantee a High Degree of Safety' generates a new situation to work on, namely the context <[Guarantee High Degree of Safety]; Refine Goal> that will be the subject of the next spiral turn.

However, this is not the only possibility. If a record of previously encountered contexts is kept then the "drawback" context (section 4) can be selected. As a consequence, the next turn of the spiral shall result in backtracking to an earlier step.

5.3: Progression in the spiral

This example shows that the progression along the spiral, from one turn to another one is based on *context-dependency.* Figure 8 extends the example to illustrate how the process can proceed from turn to turn through context dependency.

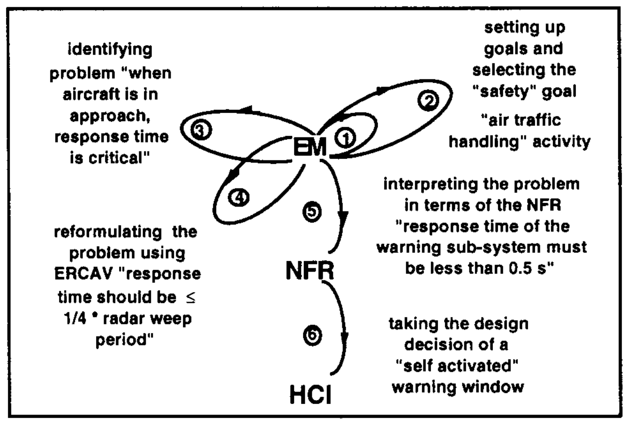

Figure 8 : Example of context dependency

Turn1 starts with the creation of the activity-typed element in the Enterprise Model (EM) called ' Air-Traffic-Handling' ;

Turn2 handles the situation raised by turn1 and focusses on a better understanding of the activity. The decision 'Improve-Activity' is explored as described in the previous section and results in the creation of the new goal-typed element in the Enterprise Model : 'Guarantee a High Degree of Safety'. Understanding the goal leads to the identification of a new problem-typed element in the EM called 'When Aircraft is in Approach Response Time is Critical'. This is performed in turn3.

The F3 set of tools includes an expert system aiming at elicitating user requirements through cooperative work. In turn4 the so-called ERCAV(Early Requirements Capture Analysis and validation) tool will be used to reformulating the problem stated in turn3 into 'Response Time should be Less than 1/4 radar weep period'.

Thus, the output of turn4 is a new element in the Enterprise Model (EM) which suggests the decision to interpret the statement as a non-functional requirement (NFR) for the system to build. The result of this interpretation is the following non-functional requirement 'Response Time should be less than 0.5 second' to be added as a new element of the Non Functional Requirements Model of the F3 product. This occurs in turn5.

Combining the decision to interpret the non-functional requirement in Human Computer Interface (HCI) terms to the situation consisting of the non-functional requirement itself defines the new context to be handled in turn6. The output of this turn is the design decision affecting the Human Communication Interface : 'The Window will be a Self-Activated Warning Window'.

Guidance of the context flow is provided by a cluster associated, in this example, to the goal "Interpret EM element in terms of IS functions" which gathers the five following chunks: <[], Identify-Activity>, <[Activity], Improve-Activity>, <[Goal], Refine-Goal>, <[Problem], Elicitate-Problem>, <[Problem], Interpret-Problem>, <[NFR], Convert into IS function>. Clustering in this case is based on dependency over contexts fulfilling the same goal.

In summary, within a turn of the spiral the requirements engineer is supported by the associated *chunk*. From one turn to another he is advised by a *chunk cluster* grouping and ordering chunks which are dependent one from the others. However the engineer has the freedom to select any other context than the one suggested by the guidance. Similarly he can break the guidance which is provided within a turn by the chunk associated to the input context.

6: Conclusion

This paper focusses on process guidance during the requirements engineering phase of Information Systems development. Guidance is examined from a double perspective : the modeling perspective and the Case perspective.

From the modeling side, the paper describes a levelwise approach in which a process model results of the instantiation of a generic meta-model for guidance and in which process traces are, in turn, instances of the process model. Therefore properties of processes and process models are inherited from the ones of the process meta-model. Guidance is, first of all, introduced in the process meta-model through the notion of context coupling a situation and a decision, objected to or supported by arguments. This allows to deal with a large number of situations in a flexible, decision-oriented manner. According to the situation the requirements engineer is faced to, he will be advised on the possible decisions to take and their evaluation criteria. In addition to this local guidance, bounded to some context, the meta-model provides a global guidance driven by the notion of goal and suggesting flows of contexts achieving the same goal. This allows to control flows in a more flexible way than prescriptions of current ways-of-working.

Finally, we believe that our approach can be applied not only to engineering activities but also to quality control and management [13,27). Further work needs to be done here.

The approach has been partially implemented in the requirements engineering environment of the Nature project. Further work will on one hand, try to validate the guidance meta-model by experiments on real case studies and, on the other hand, concentrate on the formalisation of the backtracking loop from the trace to the model.

References

[1] Armenise P., Bandinelli S., Ghezzi C., Morzenti A. : "*A survey and assessment of software process representation formalisms*", International Journal of Software Engineering and Knowledge Engineering,3,3, 1993.

[2] Boehm B.W. : "*A Spiral Model of Software Development and Enhancement*"; IEEE Computer 21.

[3] Brunet J. : "*Modelling the World with Semantic Objects*", International IFIP Conference on "The Object-Oriented Approach in Information Systems", Rolland C., Moulin B., VanAssche F. (eds), North Holland (pub), Quebec, Canada 1991.

[4] Bubenko J., Rolland C., Loucopoulos P., De Antonnellis V. : "*Facilitating Fuzzy to Formal Requirements Modeling*", Proc. Int. Conference on Requirements Engineering (ICRE94), Colorado Springs, USA, 1994.

[5] Dowson M. : "*Iteration in the Software Process*"; Proc 9th Int Conf on "Software Engineering", Monterey, CA,1988.

[6] Feiler P.H, Humphrey W.S; "*Software Process Development and Enactment: Concepts and Definitions*"; Proc. 2nd Int. conf. on the Software Process, Berlin, Germany, 1993.

[7] Finkelstein A., Kramer J., Goedicke M.: "*ViewPoint Oriented Software Development*"; Proc. Conf "Le Génie Logiciel et ses Applications", Toulouse, p 337-351, 1990.

[8] Grosz G. : "*Using Artificial Intelligence Techniques to Formalize the Information System Design Process*"; Proc Int Conf on "Databases and Expert Systems Applications", Vienna, Austria 1990.

[9] Henderson-Sellers B., Edwards J.M.; "*The Object-oriented Systems Life-Cycle*"; Comm. ACM, Vol. 09, 1990.

[10] Humphrey W.S : "*Managing the Software Process*", Addison Wesley, Reading MA, 1989.

[11] Jarke M., Mylopoulos J., Schmidt J.W., Vassiliou Y.; "*DAIDA - An Environment for Evolving Information Systems*"; ACM Transcactions on Information Systems, Vol. 10, No. 1, 1992.

[12] Jarke et Al :"*An Integrated View of Representation*", to appear in proceedings 4th European Software Engineering Conference, Garmish partenkischen, 1993.

[13] Juran J.M., Gruna F.M. : "*Quality Planning and Analysis*" McGraw Hill, 1980.

[14] Loucopoulos P., Zicari R.(eds) : "*Conceptual Modeling, Database and Case*", Wiley, 1992.

[15] Lubars M., Potts C., Richter C. : "*A Review of the State of the Practice in Requirements Modeling*", Proc. Int. Symposium on "Requirements Engineering", 1993.

[16] Maiden N., Sutcliffe A. :"*Exploiting Reusable Specifications through Analogy*", Com. ACM 35, 4, 1991.

[17] Olle T.W., Hagelstein J., MacDonald I., Rolland C., Van Assche F., Verrijn Stuart A. : "*Information Systems Methodologies : A Framework for Understanding*", Addison Wesley, 1988.

[18] Peugeot C., Franckson M.: "*Specification of the Object and Process Modeling Language*", ESF Report n° D122-OPML-1.0, 1991.

[19] Potts C.: "*A Generic Model for Representing Design Methods*"; Proceed. 11th International Conference on Software Engineering, 1989.

[20] Prakash N., Rolland C., Pernici B. (eds) : IFIP International Conference on "*The IS development Processes*", North Holland(pub), Como, Italy, 1993.

[21] Rolland C.: "*Modeling the Requirements Engineering Process*", Proc. Euro-Japanese Seminar on "Conceptual Modelling and Knowledge Base V", H.Jaakkola, H. Kangassalo, T.Kitahashi, A.Markus (eds), IOS Press, 1993.

[22] Rolland C., Cauvet C. : "*ALECSI : An Expert System for Requirements Engineering*", in "Advanced Information Systems Engineering", R.Andersen, J.A.Bubenko, A.Solvberg (Eds), Springer Verlag, 1991.

[23] Schmitt J.R.: "*Product Modeling in Requirements Engineering Process Modeling*", IFIP TC8 Int. Conf. on "Information System Development Process", Prakash N., Rolland C., Pernici B. (eds.), North Holland (pub.), Como, Italy, 1993.

[24] Spanoudakis G., Constantopoulos P. :"*Similarity for Analogical Software Reuse : A Conceptual Modelling Approach*", Proceedings 5th CAISE Conference, Rolland C. (ed), Springer Verlag, 1993.

[25] Tully C (ed) Proceedings of the 4th International Software Process Workshop, ACMSIGSOFT Sotware Engineering Notes, 14,4,1989.

[26] Royce W.W.: "*Managing the Development of Large Software Systems*"; Proc. IEEE WESCON 08/1970.

[27] Willborn W. : "*Quality Management System : A Planning and Auditing Guide*", 1989.