Requirements Eng (2002) 6:237–251

Requirements Engineering

Goal-Oriented Requirements Engineering: A Unifying Framework

Evangelia Kavakli*
Department of Cultural Technology and Communication, University of the Aegean, Mytilene, Greece

The study of contemporary requirements engineering (RE) methodologies indicates that modelling of organisational goals constitutes a central activity of the RE process. In particular, goals provide the rationale and drive the elaboration of the requirements that operationalise them. They also provide the criteria against which the completeness and correctness of the requirements specification is validated. In other words, requirements implement goals in the same way that programs implement design specifications. Despite the significance of goals in RE, research in the field is fragmented. No research has so far taken place in order to define the overall role that goals play in RE. This paper puts forward a unifying view of goal analysis in the context of RE. This allows the identification of similarities and differences between the different conceptions of goal used by different approaches and promotes the understanding of the overall role of goal analysis in RE. Based on this understanding the various approaches can be put together, thus leading to a stronger goal-driven RE framework that takes advantage of the contributions from the many streams of goal-oriented research.

Keywords: Goal analysis; Goal modelling; Goal-oriented requirements engineering; Method integration

1. Introduction

The influence of goal orientation on contemporary requirements engineering (RE) methods and techniques is evident [1]. A large number of RE approaches use the notion of goal as a high-level abstraction medium for structuring and abstracting the content of requirements [2–8]. In addition, goals are an important component of use cases in object-oriented approaches (e.g., UML [9,10]) and have also been proposed as a way to structure use cases [11]. Moreover, goal analysis is incorporated in existing methodologies. For example, the ProcessTeam[1] toolset for modelling business and information technology (IT) requirements has introduced goal modelling as part of its business definition and software implementation methodology. This widespread adoption of goal concepts in many RE approaches indicates that *goals are a core concept for RE* in general.

Existing RE frameworks (e.g., [12–14]) mention goals in several RE contexts; however, none considers the overall role of goals in RE. Moreover, authors of goal-oriented methods have convincingly argued for the significance and usefulness of goals in their respective approaches; nevertheless, there has not been a comprehensive attempt at understanding and clarifying the role of goal modelling across different stages of RE.

Different RE stages require different modelling concepts and reasoning support. Indeed, it has been argued that distinguishing the needs of early versus late RE stages can lead to conceptions of goals and goal analysis methods that are substantially different [15,16]. During the earlier stages of the RE process it is more important to model and analyse stakeholder needs and interests and how they might be addressed or compromised by the decision to introduce a new system. Later stages concern future objectives and how these may be operationalised in terms of system components. While earlier stages are characterised by uncertainty, ambiguity and value conflict, later stages focus on achieving completeness, consistency and

*Part of this work was completed while the author was a member of the UMIST Information Systems Engineering Group.

Correspondence and offprint requests to: Evangelia Kavakli, Department of Cultural Technology and Communication, University of the Aegean, Sapfous & Arionos Str, GR 811 00 Mytilene, Greece. Email: kavakli@ct.aegean.gr

[1]ProcessTeam is a trademark of Sterling Software.

precision, moving towards the final specification document.

Therefore, it is possible to make a distinction between different goal-modelling activities, which reason about different types of goals at the RE level. This fact can be established if we look at different goal-oriented approaches proposed for RE. Indeed, there are approaches (such as ISAC [17]) which emphasise the modelling of change goals, i.e. the need for introducing a system in an organisation, while other approaches (e.g., KAOS [4]) focus on the modelling of the future goals associated with the introduction of software within an organisation.

Since different goal-oriented approaches are appropriate in different RE stages, then we can argue that by putting together the various goal-oriented approaches a stronger RE framework that takes advantage of the contributions from the many streams of goal-oriented research can be built, and this is the subject matter of this paper.

In attempting to reconcile goal-oriented approaches in RE, this paper addresses the following research issues:

1. What types of goal concepts are dealt within RE?
2. What types of reasoning are associated with what conception of goal?
3. In dealing with multiple goal-oriented approaches, what approach to coupling or integration is appropriate?

To this end, Section 2 discusses the role of goals in RE. Section 3 describes a conceptual modelling technique for describing goal-oriented methods and proposes a goal-oriented method metamodel that enables modular expression of different goal-oriented methods. This modular expression is a first step towards the integration of different goal-oriented modules. In section 4, the applicability of the metamodel is demonstrated using seven different goal-oriented approaches. The alternative ways to coupling goal-oriented approaches and the associated advantages are discussed in Section 5. Section 6 places the work presented in this paper in the context of related work in the area. Finally, Section 7 concludes this paper with pointers to future work.

2. Goals in RE

Despite the fact that there is no common definition of the RE process, three tasks to be performed have been identified [18]:

- requirements elicitation;
- requirements specification; and
- requirements validation.

Requirements *elicitation* focuses on understanding the organisational situation that the system under consideration aims to improve and on describing the needs and constraints concerning the system under development. Requirements *specification* maps real-world needs onto a requirements model. Finally, the *validation* task intends to ensure that the derived specification corresponds to the original stakeholder needs and conforms to the internal and/or external constraints set by the enterprise and its environment.

Examination of current research has highlighted a large number of cases where goal analysis techniques have proven successful in the context of the different RE tasks.

In the context of requirements elicitation many RE approaches, implicitly or explicitly, represent the goals of individuals, groups or organisations, in order to describe current organisational behaviour.

For example, in Dobson et al. [5] goals appear in the guise of *policies* expressed implicitly in behavioural terms, and in the guise of an individual's *responsibility for behaviour* – an individual is responsible to some other individual, for some state of affairs. In the goal-based workflow approach [19] an organisation is seen as a tuple [`G`, `A`, `R`], where `G` is a set of goals, `A` is a set of actors and `R` is a set of resources. Actors act collaboratively using resources in order to attain their goals. Also the *i** approach [8] describes work organisation in terms of *strategic dependencies* among actors. An actor is an active entity that carries out actions to achieve goals. These *goals* are made specific, embedded in the dependencies between actors.

Another line of work in this area uses goal analysis in order to understand the need for changing the existing situation. The Information Systems Work & Analysis of Changes (ISAC) [17] uses goal analysis in order to ensure that the business problems to be solved are identified and that these problems are diagnosed correctly. To this end, ISAC supports the identification of business goals and problems that inhibit the achievement of these goals. Similarly, in the F^3 framework [3] the need for change is expressed in terms of the enterprise goals along with the current problems and weaknesses obstructing goals achievement, as well as the future threats or opportunities suggesting new goals. The Goal Directed Change (GDC) framework [20] also uses goal-modelling techniques in order to analyse the need for change and to specify alternative plans for realising this change.

In requirements specification the main objective of goal-modelling approaches has been linking business needs and objectives to system functional or non-functional components.

KAOS [4,21] is one of the most well-known software engineering approaches that stresses the importance of explicitly representing and modelling organisational goals. In KAOS requirements specification consists of progressively refining high-level goals until constraints, objects and operations that are assignable to individual agents are obtained. The GBRAM [2] method also uses goals as the means to elaborate and structure system requirements. GBRAM offers prescriptive guidelines on how to extract goals from different sources into one ordered goal set. The operationalised goals, responsible agents, stakeholders, scenarios and obstacles provide a representation of system requirements organised according to system goals. In addition, the NFR framework [22] provides for the representation of non-functional requirements in terms of interrelated goals.

Furthermore, many scenario-based approaches consider goals as a contextual property of a scenario, i.e., a property that relates the scenario to its organisational context [23]. Cockburn [11] goes beyond this view and suggests the use of goals to structure scenarios by connecting every action in a scenario to a goal assigned to an actor. In this sense a scenario is discovered each time a goal is. In a similar way, Achour et al. [24] proposes the organisation of scenarios using goal hierarchies. A goal is defined as something a stakeholder hopes to achieve in the future, while a scenario expresses a possible way in which the goal can be achieved.

Finally, in the context of requirements validation goal analysis techniques have been used to define the stakeholders' criteria against which the fitness of system components is assessed. For example, the GQM approach by Basili and Rombach [25] supports the identification of metrics from goals through the use of appropriate questions. In addition, Wilson et al. [26] uses modelling of goals in order to explicitly link safety goals to analysis results, in the context of safety-critical application design.

The role of goal modelling in relation to the three RE activities is summarised in Table 1. As can be seen in Table 1, different RE tasks require reasoning about different types of goals. In particular, during requirements elicitation one needs to reason about the current organisational goals and how these are realised in existing system components. In addition, during requirements elicitation we need to understand the motivation for changing the current situation (i.e., we need to model the change goals). In contrast, in requirements specification the focus is on future business goals and how these can be operationalised into system components. Finally, during the validation of system requirements the focus is on the stakeholders' evaluation goals and how the derived specification conforms to these goals. Therefore, we can differentiate between four types of goals at the RE level, namely: *current goals*, *change goals*, *future goals* and *evaluation goals*.

Table 1 also indicates that *research in the area is fragmented*. Coverage of the area tends to focus each time on specific RE issues, while no research has so far taken place in order to define the overall role that goals play in RE. Furthermore, contributions from different frameworks seem to *complement* each other, thus by combining the various goal-based approaches together one could obtain an improved framework that brings together the benefits of the different goal-oriented approaches.

3. A Conceptual Framework for Unifying Goal-Oriented RE Approaches

In search for a framework for unifying goal-oriented RE approaches one could start from conceptual frameworks that have been developed for characterising and understanding the relationships among the various activities, issues and aspects of RE.

For example, in Pohl [12] the RE process is viewed as progressing along three dimensions: specification, representation and agreement; Sutcliffe [13] considered RE from the viewpoint of task activities, initiating conditions and several product dimensions; Zave [14] used two major dimensions – problems and contributions to solutions – to classify research efforts in RE. While goals are mentioned quite prominently in all three frameworks, none offers any immediate suggestions on how goals might have an overall role in RE [27].

Table 1. The role of goal-analysis in relation to RE activities

RE activity	Goal analysis contribution	Goal-oriented approach
Requirements elicitation	1. Understanding the current organisational situation,	ORDIT, *i**, [19]
	2. Understanding the need for change	ISAC, F^3, GDC
Requirements specification	3. Relating business goals to functional and non-functional system components	KAOS, GBRAM , the NFR framework, [11,24]
Validation	4. Validating system specifications against stakeholders' goals	[26], GQM

In this paper we use the RE framework proposed in Kavakli and Loucopoulos [28]. This framework sees RE mainly as a knowledge-modelling process. Knowledge about the existing situation, about the alternative hypotheses on how to alter the current situation and about the possible future system all have to be modelled in some way. Knowledge models form both the result from RE tasks and the basis for reasoning during the RE process.

In particular, the framework defines the following knowledge modelling states that might be reached in an RE project (see Fig. 1):

1. The `As_Is` state concerns knowledge about the current organisational situation.
2. The `Change` state refers to the reasons (the need) for altering the existing situation through the development of the IT system under consideration.
3. the `To_Be` state characterises knowledge about the future enterprise situation.
4. The `Evaluation` state concerns knowledge regarding the assessment of the current situation, the suitability of a change plan, or the appropriateness of a future enterprise model.

The RE process can be seen as the systematic progression through the four knowledge modelling states. For example, in a reverse engineering project a possible route may start by understanding the current situation (reach the `As-Is` state) and proceed with exploring alternative change plans (reach the Change state), continuing with the evaluation of alternative plans (reach `Evaluation` state) and, finally, the design of the re-engineered business processes according to the selected change plan (reach the `To-Be` state). Alternatively, a software engineering project may start with the decision to introduce a particular information system (from the `Change` state) then proceed to define the future goals of the new composite system and the way these may be operationalised (reach the `To-Be` state) and, finally, evaluate alternative operationalisations (reach the `Evaluation` state).

Thus, there is no particular ordering between these states; i.e., there is no unique *route* for navigating the space determined by the four knowledge-modelling states. Instead, each state to be reached is dynamically selected in the course of the RE process. The sequencing of states as well as the particular manner of reaching each state is not prescribed but depends on the enactment context of the particular RE project. Alternative routes correspond to alternative ways-of-working.

From a goal-oriented perspective, each knowledge state can be described in terms of appropriate goal models. Indeed, the `As-Is` state can be represented in terms of the *current organisational goals* and the way these are achieved through the current enterprise behaviour. The `Change` state is described by *change goals* and the way these are satisfied in terms of alternative change plans. In a similar way the `To-Be` state can be seen as the model of the *future enterprise goals* and the way these are operationalised in terms of the re-engineered enterprise behaviour. Finally, the `Evaluation` state can be described in terms of the stakeholders' *evaluation goals*.

Depending on their adopted way-of-working, different goal-oriented approaches suggest alternative ways of progressing from one state to another based on reasoning about the corresponding type of goal.

Different methods emphasise and are therefore stronger in reasoning about certain types of organisational goals. For example, ISAC focuses on the need for

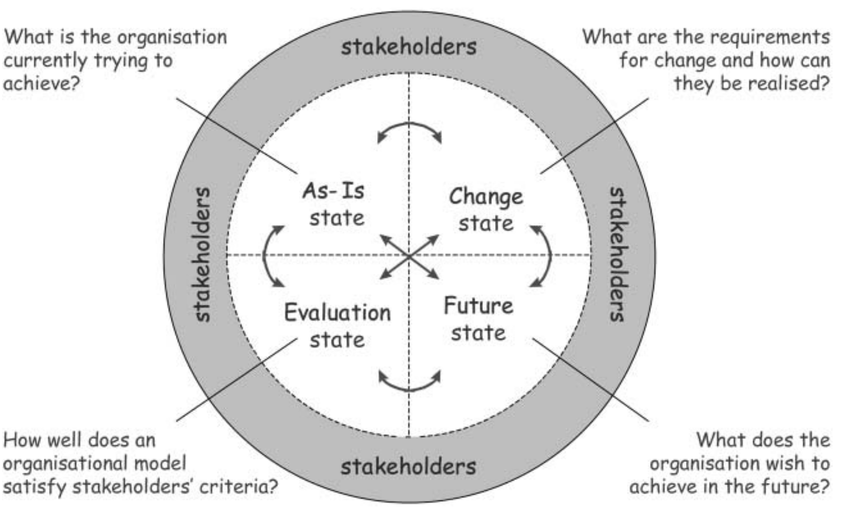

Fig. 1. The four RE knowledge-modelling states.

identifying and reasoning about change goals, while it is weak on reasoning about future goals. KAOS, in contrast, assumes a given knowledge about change goals and focus on techniques for reasoning about future goals. Furthermore, even if two goal-oriented methods focus on the same type of goals they suggest alternative strategies for reasoning about these goals. The applicability of each strategy depends on the characteristics of the particular RE project and its context. For example, strategies that prescribe stakeholder participation are efficient when dealing with uncertain situations; however, they may be difficult to apply when the heterogeneity of participants is high.

Hence, it becomes apparent that in order to enhance the effect of goal-oriented RE we need to be able to combine aspects of different goal-oriented methods (called *method fragments*), thus assembling new goal-oriented methods that best fit the needs of a particular RE project.

3.1. A Metamodelling Technique for Describing Goal-Oriented Methods

In order to assemble goal-oriented method fragments into a meaningful method, we need a suitable mechanism for modelling method fragments and providing structure to the method assembly process. These concerns are addressed through the use of a metamodelling technique comprising three abstraction levels, each dealing with different modelling scopes.

As illustrated in Fig. 2, a *method metamodel* at the *generic* level provides a set of generic concepts for expressing and composing any method. At the *method* level a *method model* defines the way-of-working related to a specific method. Finally, the actual way-of-working in a particular RE project is determined at the *project* level.

Due to the different levels of abstraction, a hierarchy of models exists, called the *genericity* hierarchy [12]. Each level within the genericity hierarchy constitutes an

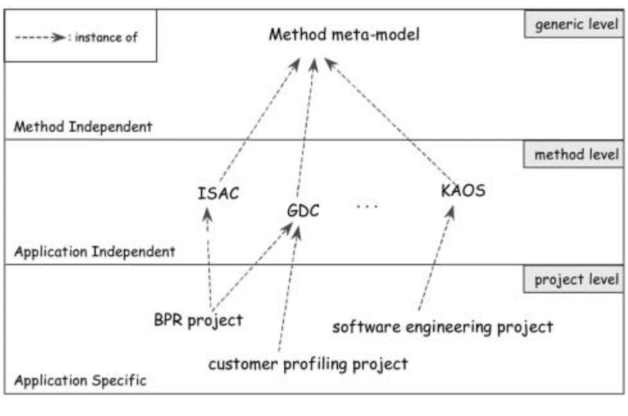

Fig. 2. Different levels of abstraction in method modelling.

instantiation of the level above and therefore has to respect its rules. Thus a method model is an instantiation of a method metamodel. Obviously a method metamodel can be instantiated several times in order to provide various method models corresponding to different methods. In a similar manner a method model may be instantiated several times through different applications of a method in different projects.

As illustrated in Fig. 2, the particular method applied in a BPR project will be described at the project level. The underlying method model that prescribes how the project should proceed according to some method (e.g., GDC) will be described at the method level. Finally, the set of generic concepts for representing the GDC method will be described at the generic level.

However, it is not compulsory that a project has to follow a specific method. Indeed a process followed in a particular project may combine aspects of different methods described in different method models.

The organisation of method information in different levels of abstraction presents a number of advantages:

- The exploitation of the metamodel allows the definition of a wide range of method models.
- The instantiation mechanism makes the activity of defining method models systematic and versatile.
- It allows for combination of different methods at the project level.

3.2. The Goal-Oriented RE Method Metamodel

As discussed earlier, a goal-oriented RE method describes a particular route that can be followed in order to reach certain knowledge-modelling states. Each route comprises a set of method-specific intentions (tasks) that have to be satisfied in order to solve a particular problem (e.g., acquire software system requirements, derive a change plan for transforming business processes).

For example, let us consider the GDC method [20]. The overall problem addressed by GDC is how to derive a satisficing plan for realising organisational change. GDC does not assume a given knowledge of the existing organisational situation; thus it suggests a reverse engineering route, illustrated in Fig. 3.

This route comprises the following tasks:

1. Discover the current enterprise situation (reach the `As-Is` state).
2. Define change needs and develop alternative change plans (i.e., reach the `Change` state).
3. Evaluate alternative plans (i.e., reach the `Evaluation` state).

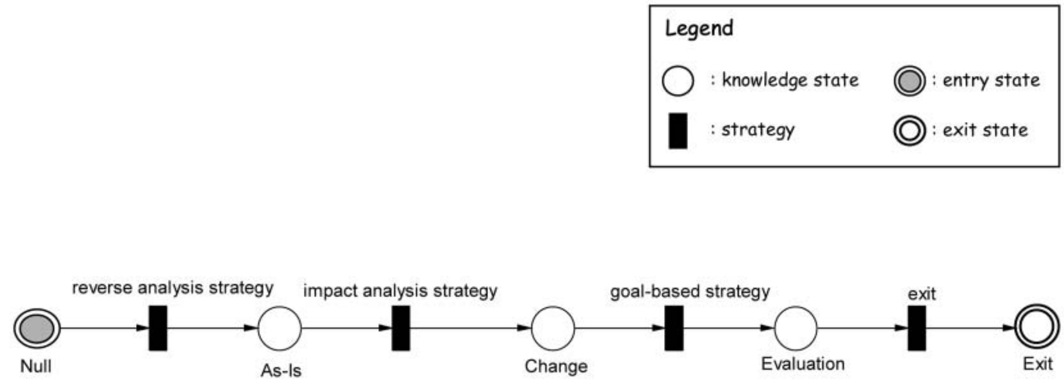

Fig. 3. Suggested route describing the GDC way-of-working.

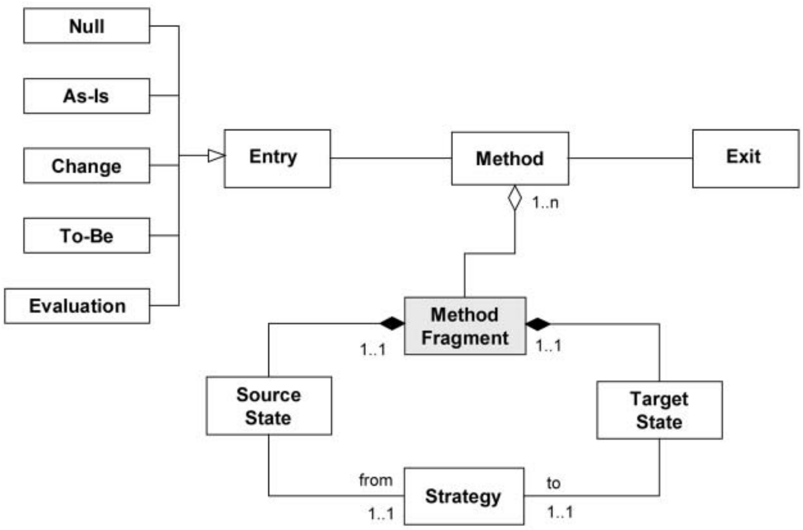

Fig. 4. The method metamodel.[3]

Each task describes the transition from an initial state to a target state. In addition to the As-Is, Change and Evaluation states that have been described in Section 3, two additional states are applicable: the Null state and the Exit state. These describe the state where no knowledge about the organisation is available and the state that 'enough' knowledge has been obtained, respectively.

The manner of performing each transition is embedded in the method in the form of method-specific *strategies*. For example, the transition from the Null state to the As-Is state according to GDC is achieved following a *reverse analysis strategy*, whereby current enterprise objectives are abstracted from existing business processes, in a bottom-up manner. The sequencing of transitions between knowledge states as well as the particular manner of performing them depends on the method followed. A specific way of reaching a target state S_j starting from an initial state S_i constitutes a strategy $\mathbf{Str_{ij}}$.[2]

There are two distinct states called Entry and Exit that represent the entry and exit states of a method respectively. Thus, a method model describes a transition from the Entry state to the Exit state. Any of the five states Null, As-Is, To-Be, Change and Evaluation can be a method Entry state. Each triplet $<S_i, S_j, \mathbf{Str_{ij}}>$ constitutes a *method fragment* (MF) and denotes the *intention* to reach target state S_j

[2]The strategy exit is used for the transition from any source state to the Exit state denoting the intention to terminate the application of the method. In most methods this transition does not involve any specific tasks. If a method prescribes a specific method strategy then this will be explicitly noted in the method's model.

[3]The concepts Entry, Exit, Source State and Target State are specialisations of the Knowledge State concept (not shown in this figure). The concept of **Strategy** is an objectified relationship that denotes the transition from the source to the target state.

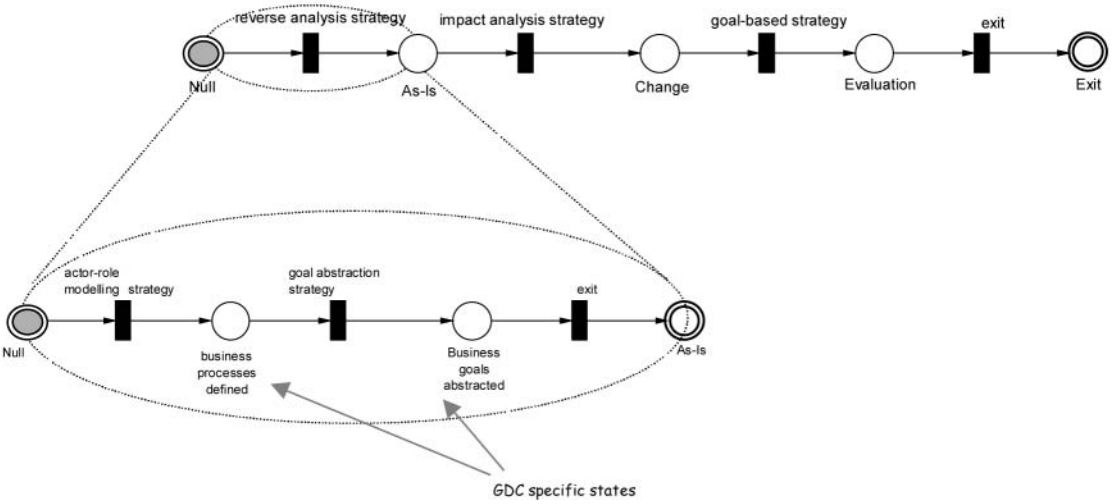

Fig. 5. Elaboration of the GDC 'reverse analysis' strategy.

from source state S_i employing a particular strategy $\mathbf{Str_{ij}}$. The above are summarised in the method metamodel of Fig. 4.

A method is visualised as a directed, bipartite graph in the spirit of Petri nets [29,30], whereby places in the net (represented by circles) correspond to knowledge states while transitions (represented by rectangles) correspond to strategies (see Fig. 3). The directed nature of the graph indicates the way one might progress from one knowledge state to the next. Using this notation a method fragment is enabled when a token (represented by a dark circle) is placed to its source state, while application of a method fragment adds a token to the corresponding target state.

Two method fragments <S_i, S_j, $\mathbf{Str_{ij}}$> and <S_j, S_k, $\mathbf{Str_{jk}}$> whereby the target state of the former is the initial state of the latter are interconnected in the method. Thus, S_k is reachable from S_i through the intermediate state S_j. For example, in Fig. 3 the method fragments <Null, As-Is, **reverse analysis strategy**> and <As-Is, Change, **impact analysis strategy**> are interconnected. Hence, the Change state is reachable from the Null state through the intermediate state As-Is.

A strategy $\mathbf{Str_{ij}}$ describes the means to carry out the transition from a source state S_i to a target state S_{ij}. Each strategy *prescribes* a method-specific way-of-working, structured according to a systematic ordering of activities. A strategy may define a number of subtasks and intermediate method-specific states that have to be reached in the context of the strategy.

For example, the reverse analysis strategy applied for the transition from the Null to the As-Is state according to the GDC way-of-working defines two subtasks: (a) model enterprise processes in terms of cooperating enterprise actors and (b) abstract enterprise goals from enterprise processes. For each subtask corresponding substrategies are provided by the GDC method (see Fig. 5). As can be seen in Fig. 5, a complex strategy can be described as a method with Entry state the source state of the corresponding fragment and Exit state the target state of the corresponding method fragment.

4. Describing Method-Specific Ways-of-Working

In this section we use the method metamodel to describe the ways-of-working of seven goal-oriented methods, namely:

- the goal-driven change method (GDC) [20];
- the ISAC change analysis [17];
- the *i** strategic rationale modelling [8];
- the NFR framework [22];
- the GBRAM goal-based requirements analysis [2];
- the goal–scenario coupling method [24]; and
- the KAOS goal-directed requirements elaboration method [4].

The objective of this analysis is to describe in a unifying manner how different well-known goal-driven approaches deal with different RE problems. Since most approaches do not explicitly define a specific way-of-working the analysis of different goal-driven approaches is mostly based on their application as found in the literature.[4]

[4]This could mean that alternative ways of working for each approach might be possible.

The first four methods (GDC, ISAC, *i** and the NFR framework) focus on 'early' stages of RE in that they emphasise the need to understand the current enterprise state and analyse the need for change from different perspectives. It should be noted that this analysis could lead to a solution that does not involve the development of a software system. In contrast, the other three approaches (KAOS, GBRAM and goal–scenario coupling) emphasise the 'later' stages of RE assuming adequate knowledge about the current organisational state and the issues of change and focus on defining the 'form' of the future solution that addresses these issues.

In the following, Sections 4.1.1 to 4.1.4 briefly discuss the particular way-of-working of each of the above methods, providing an overview of:

1. the rationale of the proposed way-of-working (what is the RE problem solved by the method, the entry and exit states and why the specific route is followed);
2. the route that is followed within the space determined by the four RE knowledge modelling states; and
3. the strategies applied.

4.1. GDC

The overall GDC objective is to assist organisational stakeholders to consciously develop schemes for introducing changes. The GDC method is based on the premise that understanding change issues depends on the knowledge shared by organisational stakeholders about the existing enterprise situation as well as the need for change. Using this knowledge, organisational change plans are systematically devised by analysing the impact of change goals on the existing situation.

The GDC way-of-working can be described by the route illustrated in Fig. 6.

Because in many real cases knowledge about the current enterprise situation is not documented, the GDC route starts from the `Null` state and proceeds by discovering and documenting knowledge about the `As-Is` enterprise state. This is based on reverse analysis of the current enterprise processes. According to this strategy, models of the current enterprise processes are developed with the help of business experts and are consequently analysed in order to identify current enterprise objectives.

The GDC route proceeds by reaching the `Change` state following an impact analysis strategy. Using this strategy, stakeholders in extensive stakeholder workshops cooperatively explore multiple change issues and how they might affect the existing enterprise structures and processes. The analysis results both in the identification of possible improvements of the current enterprise structure as well as introduction of innovative solutions.

Finally, the route terminates with the evaluation of the alternative change plans (i.e., it reaches the `Evaluation` state) using a goal-based strategy, whereby organisational stakeholders assess the appropriateness of each option against a number of applicable evaluation goals and metrics.

4.2. ISAC

The purpose of ISAC change analysis is to ensure that the business problems to be solved are identified and that these problems are diagnosed correctly. The ISAC way-of-working is based on a general problem-solving method during which current business problems are analysed, different possible solutions are investigated and one solution is chosen. The suggested 'problem-solving' route is illustrated in Fig. 7.

The problem-solving route starts with an organisational problem analysis strategy that aims to assist stakeholders in reaching an understanding of the problems that the organisation is facing in its current setting. This strategy is characterised by a high degree of participation and cooperation of stakeholders. It involves the identification and analysis of current problems and

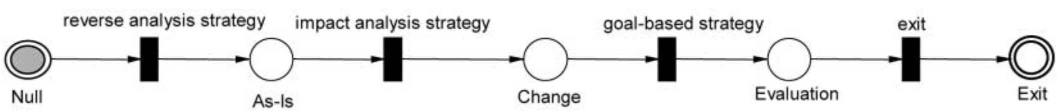

Fig. 6. The GDC way-of-working.

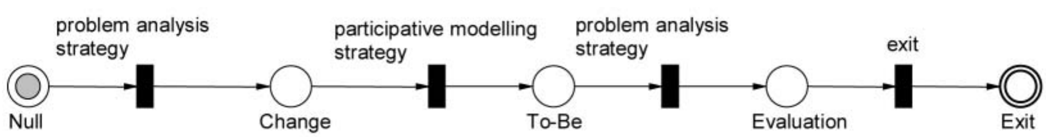

Fig. 7. The ISAC way-of-working.

the naming of the stakeholders affected (the problem-owners). Subsequent discussion of the problems with the problem owners in cooperative sessions reveals the organisational change goals as well as alternative ways to act in order to resolve current enterprise problems and reach future goals (thus reaching the `Change` state). Definition of alternative change options is facilitated by creative brainstorming sessions.

In the next step, stakeholders elaborate on the available change options and identify the changes that should be performed on existing organisational activities and create models of the future enterprise activities according to the change plan (reach the `To-Be` state). This approach is referred to as participative modelling strategy.

Finally, each of the future models generated is assessed in terms of its problem-solving power, using once again a problem analysis strategy (`Evaluation` state). In this step stakeholders investigate which of the discussed problems is addressed by the future solution and whether the introduction of this solution introduces any new problems. The most appropriate solution is the one that better addresses current enterprise problems without creating additional problems.

4.3. *i**

In the *i** method, organisational change is seen as a process of raising and resolving *strategic issues* regarding the appropriateness of work processes in the existing organisational setting. Strategic issues refer to the ability of organisational agents to achieve their goals through their involvement in organisational processes.

The *i** way-of-working suggests the following steps (depicted in the route of Fig. 8): (1) identifying and modelling the existing processes; (2) analysing the model for deficiencies; and (3) proposing new processes that resolve the identified deficiencies.

In more detail, the *i** route begins with the description of the current (`As-Is`) organisational state following the strategic actor modelling strategy. Using this technique knowledge about the current enterprise state is developed by means of modelling current enterprise actors and their rationale. The rationale of an actor refers to the individual actor's goals and how these are achieved through the actor's collaboration with other actors.

In the following step the produced organisational model is analysed in terms of the satisfiability of current goals of enterprise actors. This technique is referred to as strategic issue analysis. Satisfiability of current goals is determined based on the ability of actors to achieve their goals. The actor's ability to achieve a goal is based on the analysis of the actor behaviour as modelled using the *i** strategic rationale model. Unsatisfiable goals generate the change issues that need to be resolved by the organisational redesign process (thus, reaching the `Change` state).

Change issues are addressed in the final step resulting in new process configurations (reaching the To-Be state). Issue resolution is based on means–ends reasoning according to the strategic actor modelling strategy.

4.4. NFR

The NFR method stresses the need for developing a future solution that addresses quality requirements (termed NFR goals) such as performance, security, etc. The NFR way-of-working is shown in Fig. 9.

As seen in Fig. 9, organisational change in the NFR framework is considered an incremental process whereby future solutions are developed as an evolution of the existing situation. Thus, the first task is to develop an initial model of the existing situation (thus reaching the `As-Is` state). This is done by means of quality goal graphs. Top-level quality goals are decomposed into

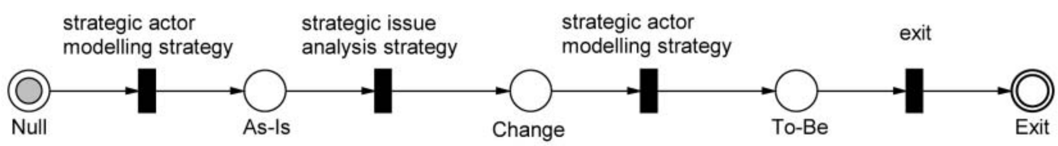

Fig. 8. The *i** way-of-working.

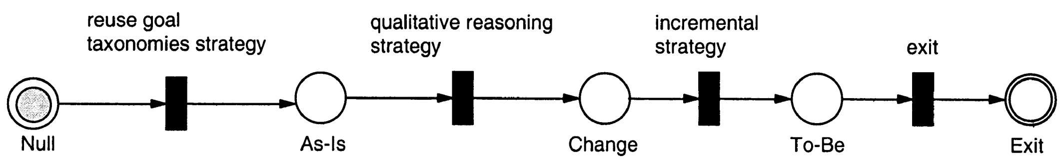

Fig. 9. The NFR way-of-working.

more specific goals until a satisficing solution is reached. The rationale behind each decomposition is also recorded in terms of corresponding arguments. In this way the initial goal graph forms a history record of the decisions that shape the current situation. Goal decomposition in each step is guided by the use of predefined quality goal taxonomies which represent specific ways for achieving quality goals. These taxonomies have been compiled from work done by researchers and practitioners in particular areas such as security and performance of information systems. Of course, depending on the situation, developers can extend or tailor the predefined taxonomies in order to accommodate the needs of the particular organisation at hand, but this knowledge reuse strategy speeds up the initial design process.

In the next step the current organisational model is analysed in terms of satisfiability. Satisfiability of the current goal graph is based on a qualitative reasoning algorithm whereby satisfiability of parent goals is calculated based on the values of offspring goals. Unsatisfiable goals as well as new organisational requirements (e.g., changes in organisational policies, reduction of staff, changes in organisational priorities) bring forward new change goals (thus reaching the `Change` state).

Change issues are addressed in the next step based on an incremental strategy; i.e., change needs are reflected in the existing quality goal graph, resulting in a revised quality goal graph (i.e., reaching the `To-Be` state). Incorporating revisions into the initial quality goal graph can be seen as deletion and/or modification of existing goals (or their relationships) as well as addition of new goals. A set of guidelines are given in order to maintain the consistency of the resulting model. Furthermore, the predefined goal taxonomies are used once more to guide the refinement of the new goals and the identification of alternative future solutions. Throughout this task design alternatives, trade-offs and decisions are again systematically recorded and can be used for further changes.

4.5. GBRAM

The objective of GBRAM is to provide practical guidance towards the identification and analysis of the future organisational goals that determine system requirements in the context of software systems development. Therefore, GBRAM concerns the later stages of RE in that the need for change has been analysed and the decision to develop some software application has been made.

The GBRAM way-of-working is shown in Fig. 10. In GBRAM discovering enterprise goals with respect to the

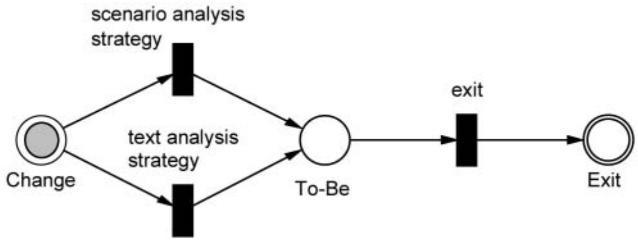

Fig. 10. The GBRAM way-of-working.

future system (i.e., reaching the `To-Be` state) is based on guided analysis of existing documents. This approach is termed text analysis strategy. Such documents may describe enterprise policies, requirement specifications of information systems as well as transcripts of interviews with stakeholders. The identification of goals is guided by heuristic rules (e.g., searching for keywords such as ‘provide’ and ‘supply’ assists the identification of goals that suggest a continual state within the organisation), and questions (e.g., ‘what goal(s) does this statement exemplify?’ and/or ‘what goal(s) does this statement block or obstruct?’).

In addition, GBRAM suggests a scenario analysis strategy based on analysis of ‘problematic’ scenarios that describe the circumstances in which a goal may fail or be blocked, thus leading to the design of solutions that resolve these problems. Scenarios are useful means for communicating with stakeholders, offering a natural way to illustrate how user needs may be satisfied or hindered in a future situation.

4.6. Goal–Scenario Coupling

The goal–scenario coupling approach assists the identification of future solutions that satisfy the organisation’s needs for change. To this end, it uses scenarios in order to elicit future organisational goals and to operationalise them in terms of system components. The corresponding way-of-working constitutes the route of Fig. 11.

According to the goal–scenario coupling strategy, identification of alternative solutions is addressed through analysis of possible future scenarios by business experts. For each future goal a scenario is described as a possible realisation of the goal. The scenario is

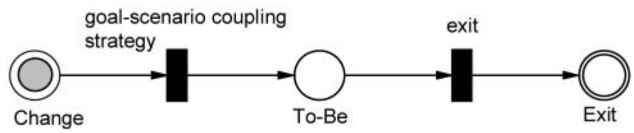

Fig. 11. The goal–scenario coupling way-of-working.

subsequently explored to yield additional future goals. The same process may be repeated for each of the discovered goals and so on. Discovery of future enterprise goals from scenarios is facilitated through the use of discovery guiding rules. For example, a goal refinement rule suggests the identification of complementary subgoals by treating every interaction between two agents in a scenario as a goal in a lower level of abstraction. The process is controlled by the scenario author (i.e., the enterprise expert), who has the authority to select among the discovered goals.

4.7 KAOS

The KAOS method also focuses on the later RE stages. In contrast to the methods described so far, which use qualitative techniques such as scenario analysis to identify and operationalise future organisational goals, KAOS focuses on a formal reasoning approach. The objective of this approach is to enable requirement engineers to automatically derive requirements specifications that satisfy the goals specified, as well as to formally verify that goals are achieved.

To this end, KAOS adopts a formal elaboration approach based on the use of domain-independent goal patterns. The suggested route is shown in Fig. 12. As can be seen in Fig. 12, definition and analysis of future goals (i.e., reaching the `To-Be` state) are based on the reuse of generic refinement patterns. These patterns are used to guide the identification and analysis of future enterprise goals as well as the operationalisation of future goals in terms of operational system components. The selection of the appropriate patterns is assisted by a number of tactics; i.e., a pattern is applicable in the context described by the tactics.

The appropriateness of a solution is subsequently validated against *correctness* properties using the proof theory of temporal logic, thus reaching the `Evaluation` state. In addition, scenario analysis is used in order to check the completeness of a design solution by searching for missing enterprise actions and their underlying goals.

5. Discussion

The analysis of existing goal-oriented approaches using the proposed method metamodel suggests that there is a *high degree of overlapping* among different methods, in the sense that different methods provide alternative strategies for serving the same RE function (i.e., they achieve the transition between the same source and target knowledge states). Such strategies may vary in terms of *focus*, *cognitive approach* and *social approach*.

The focus of a strategy refers to the particular aspect of the problem that the strategy aims to resolve. Even when two strategies tackle the same problem (the transition between the same knowledge states), they can emphasise different aspects of the problem. For example, consider the GDC **`reverse analysis strategy`** and the *i** **`strategic actor modelling strategy`**. Both strategies address the same problem: the transition from the Null state to the `As-Is` state. However, GDC focuses on the identification of global business objectives and how these are realised through business processes, while *i** focuses on goals of individual actors and how these are satisfied through the actors' collaboration.

The cognitive approach of a strategy concerns the way in which organisational knowledge is obtained and processed. We differentiate between *prescriptive*, *descriptive*, *explorative* and *analytical* approaches. In a prescriptive approach knowledge about a situation is founded on the prescriptions given by organisational experts, whereas in a descriptive strategy knowledge about the organisational situation is based on observation of the actual organisational state. In an explorative strategy knowledge is made 'visible' by means of experimentation, or systematic investigation of alternative perspectives, issues, etc. that characterise the particular organisational situation. Finally, in an analytical strategy results are deduced by reasoning on existing knowledge.

The social approach of a strategy refers to the way in which project actors work together during the analysis. Two options are distinguished: *expert-driven* strategy and *participative* strategy. In an expert-driven strategy,

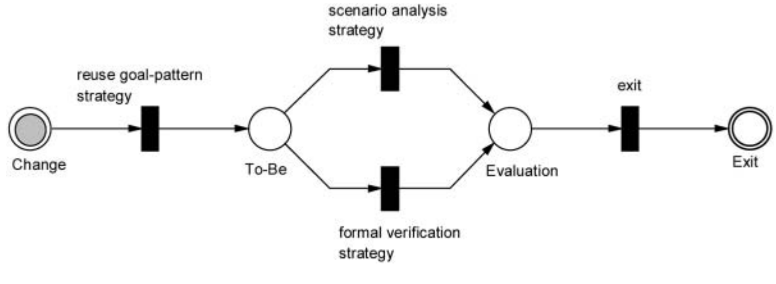

Fig. 12. The KAOS way-of-working.

the analysis is carried out by the experts in the particular technique on the basis of their own expertise, based on interviews and consultations of enterprise stakeholders. The results of the analysis can then be delivered to the stakeholders for remarks or approval. In the participative approach, experts work in close cooperation with stakeholders, e.g., in workshops with presentations, discussions, etc., with design decisions made in a cooperative manner.

Table 2 provides a summary of all strategies classified in terms of their function, focus, cognitive and social approach.

Each strategy presents different strong and weak points, depending on the nature of the particular change situation addressed. For example, prescriptive strategies are straightforward and require less effort and time for acquiring knowledge. However, they depend greatly on the experts' ability to provide correct information that is consistent with the actual situation. On the other hand, descriptive techniques best reflect how things are actually performed in the enterprise but can be more time consuming, while excessive emphasis on the current practice could divert stakeholders' attention from thinking of innovative solutions.

Analytical strategies bypass the enterprise stakeholders as a source of knowledge and focus instead on reasoning about existing information found in domain literature or constructed knowledge models. Nevertheless, the quality of the results depends on the availability and correctness of these sources. Finally, explorative approaches can be used to reduce the uncertainty of a situation by examining alternative points of view or by using various scenarios to concretise knowledge about a situation. On the downside scenarios are inherently partial; they focus on specific aspects of the problem under consideration. Furthermore, they raise the *combinatorial explosion* problem inherent to enumeration of combinations of individual behaviours [31].

Regarding the social aspects of strategies, participative approaches deal with the uncertainty inherent to change situations due to multiplicity of goals, the existence of conflicting interdependencies between goals, etc. In addition, they promote awareness of the change issues among organisational stakeholders as well as ensuring that a solution addresses the concerns of all groups affected. At the same time, participative approaches require appropriate facilitation skills and

Table 2. Overview of goal-oriented strategies

Applicable strategies	Focus	Cognitive approach	Social approach
Null→As-Is			
reverse analysis strategy	Global enterprise objectives, enterprise processes	Descriptive	Expert driven
strategic actor modelling strategy	Individual actors and their goals	Descriptive	Expert driven
reuse goal taxonomy strategy	Quality goals and how they are operationalised through enterprise components	Descriptive	Expert driven
Null→Change			
problem analysis strategy	Enterprise problems and change plans to address these problems	Prescriptive	Participative
As-Is→Change			
impact analysis strategy	Change requirements, change plans to satisfice these requirements	Descriptive	Participative
strategic issue analysis strategy	Satisfiability of current goals	Analytical	Expert driven
qualitative reasoning strategy	Satisfiability of current quality goals	Analytical	Expert driven
Change→Evaluation			
goal-based strategy	Appropriateness of alternatives with respect to contextual evaluation goals	Analytical	Participative
Change→To-Be			
participative modelling strategy	Enterprise processes	Prescriptive	Participative
strategic actor modelling strategy	Organisational actors and their goals	Prescriptive	Expert driven
incremental strategy	Future enterprise goals are how they are satisficed by alternative solutions	Analytical	Expert driven
scenario analysis strategy	Future enterprise goals and problematic scenarios	Explorative	Expert driven
text analysis strategy	Future enterprise goals	Analytical	Expert driven
goal-scenario coupling strategy	Enterprise goals and associated realisation scenarios	Explorative	Expert driven
reuse goal-pattern strategy	Future enterprise goals and how they are operationalised through enterprise components	Prescriptive	Expert driven
To-Be→Evaluation			
problem analysis strategy	Appropriateness of alternatives towards solving enterprise goals	Analytical	Expert driven
scenario analysis strategy	Completeness	Explorative	Expert driven
formal verification strategy	Correctness	Analytical	Expert driven

may be difficult to apply when the heterogeneity of participants is high.

As a result, *selection between alternative strategies cannot be prescribed in advance*; rather, a mix-and-match approach to producing a customised method for a particular project would be more effective.

An important observation is that *additional benefits can be gained by integrating different methods*. This integration can be either *function driven* or *quality driven*. In the first case, the objective is to add new functional capabilities to an existing method. This can be achieved by incorporating additional method fragments in an existing method model. For example, the addition of the KAOS method fragment <`To-Be`, `Evaluation`, **`scenario analysis strategy`**> to the *i** method model will add a further functional capability to the *i** method, that of evaluating the future organisational model using scenario analysis.

Quality-driven integration is aimed at improving the quality and usability of a method in supporting its existing features. This can be achieved by combining alternative strategies. For example, the design of the future organisational models in ISAC is based on the experience of enterprise experts. By combining this strategy with the NFR strategy, which supports the reuse of knowledge acquired in similar domains, we can reduce design effort and increase efficiency of resulting solutions. An example of quality-driven integration is found in Dubois et al. [15], whereby the *i** **`strategic actor modelling strategy`** is combined with the KAOS **`reuse goal-pattern strategy`** in order to specify the future goals of a coalmine system.

6. Related Work

The need to develop a unified view of goal-oriented approaches to RE was first addressed in Yu and Mylopoulos [27]. In their paper the authors recognise intentional concepts as part of the existential categories in the world that are of interest to the RE discipline and outline the benefits and advantages of using goal concepts. The paper offers considerable insight into the use of goals in different areas of RE but it does not offer any immediate suggestion on how to unify goal-oriented concepts and approaches in RE.

One approach is to study ontological issues directly. This approach searches for analogies and correspondences between goal-oriented approaches based on the analysis of the semantic concepts introduced by the various approaches.

Research in this direction has resulted in a number of goal classification schemes (goal taxonomies). For example, KAOS proposes two taxonomies of goals: the first is based on the formal expression of goals and distinguishes between achievement, cessation, maintenance, avoidance and optimisation goals. The second categorisation distinguishes between private goals and system goals, which are further refined into subclasses. The NFR framework uses a non-exhaustive classification of *soft* (quality) goals into (among others) performance goals, security goals, accuracy goals, whereby each goal class is refined in further subcategories. Finally, Pratt [32] approaches goal classification from a linguistic perspective and proposes a dictionary of goal verbs, built on linguistic criteria.

The purpose of the above taxonomies has been to assist a more formal expression of goals; one can thereby prove the correctness and completeness of goal analysis operations based on the semantic characteristics of each goal class. However, they offer limited assistance towards the understanding of the differences and similarities of the types of goals used in order to address different problems across the various RE activities, or of the type of analysis associated with each goal type.

In this work we approach goal categorisation from a different perspective. Instead of trying to integrate the goal categories that have been used in different approaches, we started by considering the different types of reasoning support required in different RE contexts. Analysis of existing approaches in terms of the RE stages they support revealed four general types of goals, namely current goals, future goals, change goals and evaluation goals.

This classification focuses on the product, i.e., the derived goal model resulting from the use of an approach. In addition, analysis of the operational aspects of different goal-driven approaches based on their application as found in the literature revealed a number of dynamic affinities between the different goal-driven ways-of-working.

The synthesis of the above static (product) and dynamic (process) affinities gave form to the goal-driven method metamodel that can be used to describe alternative methods. More importantly, it provides the infrastructure for modularising alternative methods in identifiable sections called method fragments. As discussed in Section 5, a method fragment is a context-sensitive series of actions; i.e., it best applies in certain design situations. However, the individual strategies and knowledge states of which it consists, are defined in a generic way; i.e., they make no reference to details particular to one application. This allows the fragments to be used in every instance of the corresponding design situation, thus enabling the integration of alternative method fragments at the project level. Subsequently, this allows the adaptation and extension of the existing goal-oriented methods so that they can fit to the characteristics of real projects and their contexts – a need stressed in recent research in the area of method engineering [33,34] and method reuse [35].

7. Conclusions and Further Work

The starting point for this work was the observation that goal modelling has been increasingly used in recent RE frameworks in order to support a number of complementary RE activities (see Table 1). Goal orientation can be seen as a fundamental movement in RE to extend its traditional ontology to encompass *intentional* concepts. Intentional concepts (of which goal is the most prominent) convey alternative visions of the world as one would desire. The consideration of goals raises the possibility of success and failure, not just truth versus falsity. This can lead to the exploration and consideration of alternatives, decision spaces and trade-offs. Very importantly, it allows the expression of freedom within such spaces; one can state a goal without having to specify how it is to be achieved [27].

The position put forward in this paper is that since intentional concepts have such fundamental significance in RE, then it is desirable to seek some coherent view of goal analysis within the field. This will allow the identification of similarities and differences between the different conceptions of goal used by different goal-oriented approaches and will shed light on our understanding of the overall role of goal analysis in RE. Furthermore, based on this understanding the various goal-oriented approaches can be put together, thus leading to a stronger goal-driven RE framework.

To this end, a goal-driven method metamodel was put forward. With this metamodel a goal-driven method can be described as a collection of method fragments, each prescribing a way of progressing from an initial knowledge modelling state to a target knowledge modelling state. The exploitation of this metamodel allows the definition of a wide range of method models and was indeed used to effectively describe seven goal-oriented methods, namely GDC, ISAC, *i**, the NFR framework, KAOS, GBRAM and goal–scenario coupling.

Furthermore, the expression of different goal-driven methods in terms of compatible method fragments can help requirement engineers to assemble fragments from different method models, thus generating new methods that better fit the needs of real projects and their contexts – a need stressed in recent research in the area of method engineering [33,34] and method reuse [35].

Method fragments can be combined in many ways, some of which can be meaningless. To this end, a form of support is necessary in order to assist requirements engineers to perform this integration. In particular, such a tool should address the following concerns:

(a) strategic (method assembly) concerns: what method fragments are applicable in a given situation and why?
(b) tactical (method application) concerns: how can a specific method fragment be applied?

Current work focuses on the definition of a 'methodology roadmap', for guiding the integration of goal-driven method fragments. Another important line of work concerns the integration of goal-driven method fragments to other non-goal-oriented approaches. Indeed, the ability to represent goal-driven methods in a modular way enables their treatment as reusable components that can be deployed in a variety of contexts, including agent-based approaches [36], or scenario-based approaches [24,37–40].

Acknowledgements. The author wishes to acknowledge the assistance of Professor Peri Loucopoulos for his ongoing help in this work. Many thanks to the anonymous reviewers for their helpful comments, as well as to Ms Shermain James for her work in guiding this paper through the publication process.

References

1. van Lamsweerde A. Requirements engineering in the year 00: a research perspective. Invited paper for ICSE'2000 – 22nd International Conference on Software Engineering, Limerick. ACM Press, New York, June 2000
2. Anton A. Goal-based requirements analysis. In: ICRE'96, IEEE, Colorado Springs, CO, 1996, pp 136–144
3. Bubenko J. 1994. The F3 Reference Manual, Deliverable F3, version 0.4, June, 1994. Department of Computer Science, University of Stockholm, Sweden
4. Dardenne A, van Lamsweerde A, Fickas S. Goal-directed requirements acquisition. Sci Comput Program 1993;20:3–50
5. Dobson JS, Blyth AJC, Chudge J, Strens R. The ORDIT approach to organisational requirements. In: Jirotka M, Goguen JA (eds). Requirements engineering: social and technical issues. Academic Press, London, 1994, pp 87–106
6. Loucopoulos P, Kavakli E. Enterprise modelling and the teleological approach to requirements engineering. Int J Intell Cooperative Inform Syst 1995;4(1):45–79
7. Ramesh B, Dhar V. Supporting systems development by capturing deliberations during requirements engineering. IEEE Trans Software Eng 1992;18(6):498–510
8. Yu E. An organisational modelling framework for multi-perspective information system design. Requirements Eng 1993 – Selected Papers, J. Mylopoulos et al. eds, Tech Rpt DKBS-TR-92-2, Dept. Comp Sci, Univ of Toronto, July 1993, pp 66–86
9. Booch G, Rumbaugh J, Jacobson I. The unified modeling language user guide. Addison-Wesley, Reading, MA, 1998
10. Jacobson I. (1995) The use-case construct in object-oriented software engineering. In: Carroll JM (ed.). Scenario-based design: envisioning work and technology in system development. Wiley, New York, 1995, pp 309–336
11. Cockburn A. Structuring use cases with goals. J Object-Oriented Program (JOOP/ROAD) 1997;10(5):56–62. http://members.aol.-com/acockburn/papers/usecases.htm
12. Pohl K. Process-centered requirements engineering. Research Studies Press, Taunton, UK, 1996

13. Sutcliffe A. A conceptual framework for requirements engineering. Requirements Eng 1996; 1(3):170–189
14. Zave P. Classification of research efforts in requirements engineering. ACM Comput Surveys 1997;29(4):315–321
15. Dubois E, Yu E, Petit M. From early to late formal requirements: a process-control case study. In: Ninth IEEE international workshop on software specification and design (IWSSD-9), Isobe, Japan, 1998
16. Yu E. Towards modelling and reasoning support for early-phase requirements engineering. IEEE international symposium on requirements engineering, IEEE, Annapolis, MD, 1997, pp 226–235
17. Lundeberg M. The ISAC approach to specification of information systems and its application to the organisation of an IFIP working conference. In: Olle TW, Sol HG, Verrijn-Stuart AA (eds). Information systems design methodologies: a comparative review. IEEE Computer Society Press, North-Holland, 1982, pp. 273–234
18. Loucopoulos P, Karakostas V. System requirements engineering. McGraw-Hill, London, 1995
19. Ellis CA, Wainer J. Goal-based models of collaboration. Collaborative Comput 1994;1:61–86
20. Kavakli E. Goal-driven requirements engineering: modelling and guidance. PhD thesis, UMIST, 1999
21. Lamsweerde A, Darimont R, Massonet P. Goal-directed elaboration of requirements for a meeting scheduler: problems and lessons learnt. In: Requirements engineering workshop, IEEE, York, UK, 1995, pp 194–203
22. Chung L, Nixon BA, Yu E. Dealing with change: an approach using non-functional requirements. Requirements Eng 1996;1(4):238–259
23. Filippidou D. Designing with scenarios: a critical review of current research and practice. Requirements Eng 1998;3(1):1–22
24. Ben Achour C, Rolland C, Souveyet C. A proposal for improving the quality of the organisation of scenario collections. In: REFSQ'98, Pisa, Italy, 1998
25. Basili VR, Rombach DD. 1988. The TAME project: towards improvement-oriented software environments. IEEE Trans Software Eng 1988;14(6):758 ff
26. Wilson SP, Kelly TP, McDermid JA. Safety case development: current practice, future prospects. In: 1st ENCRESS/5th CSR workshop, Bruges, Belgium, September 1995
27. Yu E, Mylopoulos J. Why goal-oriented requirements engineering. In: Dubois E, Opdahl A, Pohl K (eds). Fourth international workshop on requirements engineering: foundation for software quality (REFSQ'98), Pisa, Italy, 1998
28. Kavakli V, Loucopoulos P. Modelling of organisational change using an intentional approach. Commun Assoc Inform Syst (CAIS) 1999;2(6)
29. Peterson JL. Petri net theory and the modeling of systems. Prentice-Hall, Englewood Cliffs, NJ, 1981
30. Reisig W. Petri nets: an introduction. Springer, Berlin, 1985
31. van Lamsweerde A, Willemet L. Inferring declarative requirements specifications from operational scenarios. IEEE Trans Software Eng, Special issue on scenario management, 1998;24(12):1089–1114
32. Pratt N. Goal formalisation and classification for requirements engineering. In 3rd international workshop on requirements engineering: foundations of software quality (REFSQ'97), Barcelona, Spain, 1997, pp 145–156
33. Brinkkemper S. Method engineering: engineering of information systems development methods and tools. J Inform Software Technol 1996;38(4):275–280
34. Brinkkemper S, Saeki M, Harmsen F. Assembly techniques for method engineering. In: Pernici B, Thanis C (eds). CAiSE'98, Pisa, Italy. Springer, Berlin, 1998, pp 381–400
35. Rolland C, Plihon V, Ralyte J. Specifying the reuse context of scenario method chunks. In: Pernici B, Thanos C (eds). CAiSE'98, Pisa, Italy. Springer, Berlin, 1998, pp 191–218
36. Yu E. Why agent-oriented requirements engineering. In: 3rd international workshop on requirements engineering: foundations of software quality (REFSQ'97), Barcelona, Spain, 1997, pp 171–183
37. Anton AI, McCracken WM, Potts C. Goal decomposition and scenario analysis in business process reengineering. In: Wijers G, Brinkkemper S, Wasserman T (eds). 6th international conference on advanced information systems engineering (CAiSE'94), Utrecht, The Netherlands. Springer, Berlin, 1994, pp 94–104
38. Kavakli E, Loucopoulos P, Filippidou D. Using scenarios to systematically support goal-directed elaboration for information systems requirements. In: ECBS'96, IEEE Computer Society, Friedrichshafen, Germany, 1996
39. van Lamsweerde A, Willemet L. Inferring declarative requirements specifications from operational scenarios. IEEE Trans Software Eng 1998;24(12):1089–1114
40. Weidenhaupt K, Pohl K, Jarke M, Haumer P. Scenarios in system development: current practice. IEEE Software 1998;15(2)34–45