Goal-based organizational modeling oriented towards late requirements generation using the Tropos Framework

Hugo Estrada[1,2], Jaelson F.B. Castro[3], Oscar Pastor[1] ,Alicia Martínez[1,4]
[1] Technical University of Valencia
Avenida de los Naranjos s/n, Valencia, Spain
{hestrada, alimartin, opastor}@dsic.upv.es
[2] CENIDET Cuernavaca, Morelos, Mexico
[3] UFPE Recife, Pernanbuco, Brazil jbc@cin.ufpe.br
4 I.T. Zacatepec, Morelos, Mexico

Abstract. Recently, a lot of research efforts in software engineering have focused on integrating organizational modeling as a key piece in requirements engineering. In these research works, the business models are proposed as the source of the software requirements specification process. However, the majority of these works focus only on the definition of notations that permit the representation of the semantics of the organizational context, and only a few works define processes that use these notations in a methodological way. This lack of a methodological process for generating organizational models makes practical application in software development enterprises difficult. The objective of this paper is to present a goal-based method to generate organizational models represented in the Tropos framework. These models are later used as starting point of the requirements specification process of the information system. This will enable us to develop information systems that integrate the necessary functionality so that the organizational actors perform their tasks and fulfill their goals.

Keywords: Business Modeling, Goal Modeling.

1 Introduction

Traditionally, requirements engineering has been defined as the systematic process of identification and specification of the expected functions of a software system. However, this approach has certain weaknesses. McDermind [Mcd96] indicates that when the functional specification of the software system is the focal point of the requirements analysis, requirements engineers tend to establish the scope of the software system before having a clear understanding of the user's real needs. It constitutes a very important reason why many of the systems developed from a requirements model that focuses only on the functionality of the software system do not comply with their correct role within the organization.

It is important to point out that the main objective of an information system is to automate certain tasks or activities in a business process, allowing the organizational actors to reach their particular goals, as well as the general goals of the organization. In this context, there are research works that highlight the importance of using organizational models as a starting point in the development of information systems [Bub94][Ces02][Lou95][Cas02]. The Tropos Project [Cas02] has proposed a software development methodology which is inspired by organizational concepts and which reduces the impedance mismatch between an information system and its intended environment. Unfortunately, in Tropos, as well as in the majority of the research efforts in business modeling, there is still no method that creates an initial organizational model, which guides the analyst in eliciting the relevant information from the organizational context. The objective of this paper is to present a novel method to create

organizational models, represented in the Tropos Framework. The organizational models generated will then be used to generate requirement specifications [San03].

The paper is structured as follows: Section 2 describes the Tropos methodology. Section 3 presents an overview of the presented method. Section 4 presents the Case Study. Section 5 presents the Goal-Based Organizational Modeling Method proposed in this paper. Section 6 presents related work. Finally, Section 7 presents the conclusions.

2. The Tropos Methodology

Tropos proposes a software development methodology and a development framework which are based on concepts used to model early requirements. They are based on the premise that in order to build software that operates within a dynamic environment, it is necessary to analyze and explicitly model that environment in terms of actors, their goals and dependencies on other actors.

Tropos proposes five phases of software development:

- Early requirements, which is concerned with the understanding of a problem by studying an organizational setting. The output is an organizational model which includes relevant actors, their goals and dependencies.
- Late requirements, in which the system-to-be is described in its operational environment, along with relevant functions and qualities.
- Architectural design, in which the system's global architecture is defined in terms of subsystems, which are interconnected through data, controls and dependencies.
- Detail design, in which the behavior of each architectural component is defined in further detail.

In this work, we focus on the early and late requirements phases.

To support modeling and analysis during the early requirements, Tropos adopts the concepts offered by i* [Yu95], a modeling framework defined in terms of concepts such as actors and social dependencies among actors, including goal, softgoal, task and resource dependencies.

In Tropos we have the following concepts:

- Actor: An actor is an active entity that carries out actions to achieve goals by exercising its know-how.
- Dependency: A dependency describes an intentional relationship between two actors. It is composed by: a) *Depender*: the actor who is dependent on another actor, b) *Dependee*: the actor on whom another actor depends, c) *Dependum*: the task, goal, resource or softgoal on which the relationship is focused.
- Goal dependency, in which an actor depends on another actor to fulfill a goal, without prescribing the way in which it should be carried out.
- Resource dependency, in which an actor depends on another actor to deliver a resource that can be either material or informational.
- Task dependency, in which there exists a dependency for the carrying out of a task, establishing the way in which it should be performed.
- Softgoal dependency. This is similar to the goal dependency, with the difference that the goal has not been precisely defined. This dependency type does not appear in our approach, due to the fact that these types correspond to non-functional requirements, which are not considered in our proposal.

By using these elements, it is possible to create the i* Strategic Models: the Strategic Dependency Model and the Strategic Rationale Model.

The *Strategic Dependency Model* (SD) shows the dependencies that exist between actors in a business process. The model is represented by a graph where nodes represent actors, and where the links represent the dependencies that exist between these actors to achieve their goals, carry out tasks and provide or request resources.

The *Strategic Rationale Model* (SR) carries out a deeper reasoning of the motives that exist behind each dependency relationship. This is useful for representing tasks that have to be carried out by the actors to achieve the goals which are expected of them, as well as for rethinking new ways of working. This model is based on the elements of the dependency model, adding a) task decomposition links which allow us to represent the combination of necessary tasks to achieve a goal, and b) mean-ends links whose objective is to present the diverse alternatives that can be taken to fulfill a task or goal.

The Tropos methodology has been used in several application areas, including requirements engineering, software processes and business process reengineering. However, In Tropos Methodology, there is still no method that creates an initial organizational models (Early requirements acquisition), which guide the analyst to elicit the relevant information from the organizational context.

3. Overview of the proposed method

The objective of the proposed method "Goal-based Organizational Modeling" is to help to construct the Organizational Models of the Tropos Methodology using a goal-based method.

The four steps of our proposed method (Figure 1) are summarized as follows:

1. Use a Goal-Based Elicitation Method to construct a Goal-Refinement Tree (GRT) which captures the organizational context.
   1.1 Using a Goal-Refinement Strategy.
   1.2 Using a Goal-Abstraction Strategy.
2. Use the GRT to analyze the several possibilities to satisfy goals or complete operations in the organization.
3. Create the Strategic Models of the i* Framework using the GRT as starting point. The Strategic Models could then be used to perform organizational analysis to improve the model.
   a) Obtaining the Strategic Dependency Model.
   b) Obtaining the Strategic Rational Model.
4. The software system actor is later inserted inside these organizational models, and the responsibilities of the organizational actors are delegated towards the software system actor (the system to be developed). The revised organizational model can be used to derive functional (use case) specification with their corresponding scenarios [San03].

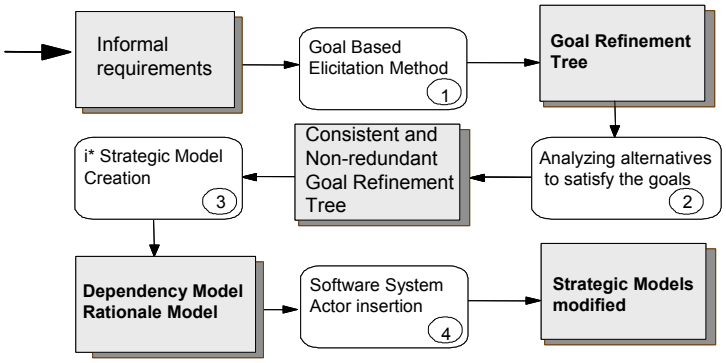

Figure 1 A fragment of the Goal-Refinement Tree obtained with the Refined Strategy

4. The Case Study

In order to illustrate our approach, we take as a case study the organization of a meeting, i.e. "*Organize a Workshop*", and we focus on the process of papers review to participate in the Workshop. The objective of this case study is to model two different alternatives that exist when organizing such technical meeting. One of them is the "quality review process". The objective of this alternative is to select the best papers and to give a quality feedback to the authors, in the other hand, we analyze the "simplified review process". The objective of this alternative is to reduce the time and effort used in the organization of the workshop. In both cases, we analyze the trade-off between the two alternatives to check if the goals of the organizational actors are fully satisfied.

5. The Goal-Based Organizational Modeling Method

In this section, we describe the steps shown in the previous section in detail. Section 5.1 shows the Goal-Based Elicitation Method. Section 5.2 shows the alternatives analysis. Section 5.3 presents the method to create the Strategic Dependency Model using the GRT. Section 5.4 presents the insertion of the software system actor. The case study us used to show the application of each step of the proposal.

5.1 Step 1: Using the Goal-Based Elicitation Method

The Goal-Based Elicitation Method proposed in this paper allows us elicit the organizational goals and to represent these in a goal structure. To do this, we propose a Goal Classification, which permits us to construct a Goal-Refinement Tree using Refinement and Abstraction Strategies.

The root of the Tree represents one of the general goals of the organization. The intermediate nodes represent the groups of low-level goals for the satisfaction of a more general goal. Finally, the leaves could represent operational goals which satisfy the low-level goals or could represent goals which will be refined in succeeding modeling phases. A more detailed Goal Classification was proposed to structure the GRT.

5.1.1 Goal Classification and Goal Structure in the GRT.

We propose a goal classification to structure the goals in the GRT. We present an example for each one of the goal types.

- **Operational Goal**: These goals can be satisfied by the correct state transition of one of the organizational actors [Dar93]. There are two types of Operational Goals:
  - Operation-Dependency. In this case, the actor responsible for completing the operation depends on another actor to provide a resource or perform another operation. These kinds of Operational Goals are represented in the GRT as OP-Dep.
  - Operation Without-Dependency. In this case, the actor responsible for completing the operation does not depend on another actor to complete the operational goal. These kinds of Operational Goals are represented in the GRT as OP-WDep.
- **Achievement Goals**: These are the goals that are refined only in Operations Without-Dependency. They are represented in the GRT as *AG*.

- **Achievement-Dependency Goal**: These are goals that are refined Operational Goals, where at least one of these is an Operations-Dependency. They are represented in the GRT as *ADG*.
- **General Goals**: These are high-level goals that are used to express the business manager's point of view. Goals of this type lead directly to General Goals, Achievement Goals or Achievement-Dependency Goals.

We have defined mechanisms to structure the Goal-Refinement Tree.

- Conflict Goals: These are goals whose satisfaction leads the actors to contradictory states. They are represented by CG.
- Decomposition Goals: These represent the necessary Subgoals to satisfy a more general goal. These are represented using the link —+ which links the subgoals with the goal.
- Alternative Goals: This is the case in which only one of the goals should be satisfied. These represent a decision structure to show the alternatives that exist to achieve a goal. They are represented using the link —► which links the alternatives subgoal with the more general goal.

5.1.2 Refinement Strategy to create the GRT

This top-down goal analysis is useful in the cases where the analyst elicits the goal of the organizational managers, who tend to express high-level goals.

In the *refinement strategy*, it is necessary to select some of the general goals of the organization and determine the subset of subgoals that permit us to satisfy it. This information is used to construct the high levels of the Goal-Refinement Tree (General Goals). It is possible to continue the refinement to detect low-level goals or operations that satisfy the high-level goals. Once the low-level goals or the operations are determined, it is necessary to find the actors responsible for accomplishing them.

Figure 2 shows a fragment of the GRT generated by the Refinement Strategy where the more general goal is "*organize a Workshop*". To achieve this goal, it is necessary to achieve the subgoals: *select papers to be presented*, *find a correct location for the Workshop, find financial aid, define a schedule,* etc. At the same time, to satisfy the goal *select paper to be presented*, it is necessary to *implement a papers review process* for the workshop.

In the refinement process, it is possible but not necessary to refine all the goals into Operation-Dependency or Operations Without-Dependency. In Tropos methodology, it is possible to leave goals to be operationalized in subsequent phases of organizational modeling. These goals are represented using a * symbol. In the case of goals that have been operationalized, it is necessary to find the actor responsible for achieving the operational goal.

5.1.3 Abstraction Strategy to create the GRT

This bottom-up goal analysis is useful in the case where the analyst elicits the goal of the organizational actors who tend to express low-level goals.

In the *abstraction strategy*, it is necessary to detect the actors who participate in the organization. Once the organizational actors are detected, their goals and operations need to be elicited. This information is used to construct the low levels of the Goal-Refinement Tree (Operational Goals). Later, it is necessary to determine the objective of the execution of the actor operations and to determine the more general goals that are satisfied by the more specific goals of the actors.

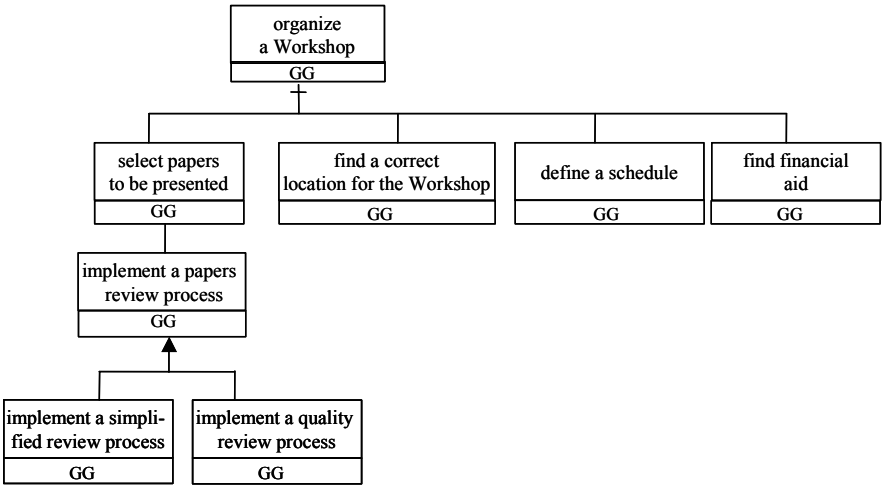

Figure 2 A fragment of the Goal-Refinement Tree obtained with the Refined Strategy

The goals of the actor must be represented in the Goal-Refinement Tree in a direct or indirect way. The specific actor goals may need to be translated into a more general goal of the organization. For example, the goals of the Authors: *have a paper accepted in the workshop* and to *obtain feedback* could be satisfied by the goal *Send Notifications and Reviews*.

In the process of identification of the actors responsible for achieving the Operational Goal, it is possible to find dependency relationships among actors. There are dependency relationships when it is necessary for another actor to provide a resource or perform another operation to satisfy a specific operation of an actor. These dependencies must be represented in the Goal-Refinement Tree as Operation-Dependency. For example, one of the goals of the PcChair is to *Obtain a large number of quality papers*; however, to achieve this goal, the PcChair depends on the Authors to submit their papers to the workshop. In this case, it is necessary to create an Operation-Dependency called *Obtain papers*.

In the Abstraction Strategy, as well as in the Refinement Strategy, it is possible but not necessary to refine all the goals into Operation-Dependency or Operations Without-Dependency. Figure 3 shows a fragment of the GRT generated by the Abstraction Strategy.

5.1.4 Result of the goal-based elicitation process

As result of the Abstraction and Refinement process, we determine that it is possible to have two different alternatives to satisfy the goal *Implement a papers review process*: *implement a simplified review process* and *implement a quality review process*. In both, the goal *select paper to be presented* is achieved.

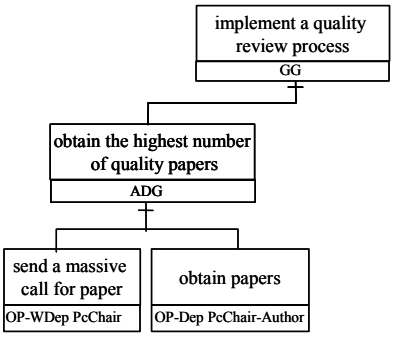

Figure 3 A fragment of the GRT obtained with the Abstraction Strategy

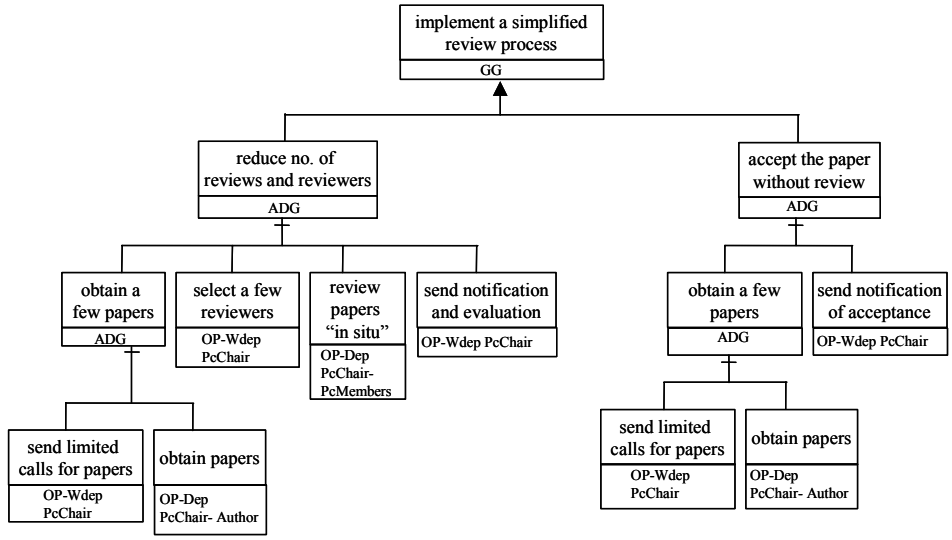

Figure 4 The GRT of the alternative: *Implement a simplified review process*

Figure 4 shows the GRT for the case *implement a simplified review process*. In this case, the strategy to satisfy the goal consists of reducing the number of reviews or of eliminating the reviews and accepting all the papers if the number of papers is limited. This last alternative was a real case of a small workshop of an important conference. In this case, due to the few papers submitted by the authors, the PcChair decided to accept all the papers without reviews and to send the acceptance notifications to the authors. Therefore, this must be analyzed as this goal model has repercussions in the satisfaction of the goals of the organizational actor.

Figure 5 shows the GRT of the case *implement a quality review process*. In this case, the strategy to satisfy the goal consists in selecting adequate PcMembers and Reviewers to review the paper. The PcChair performs the paper assignations based on the topics of interest of the PcMembers. The PcMembers are responsible for assigning the papers to the Reviewers. Finally, with the reviews, the PcChair creates a list of accepted and rejected papers and sends the notifications to the authors. As in the case of *implement a simplified review process*, it is necessary to evaluate the repercussions of this goal model on the satisfaction of the goals of the organizational actors.

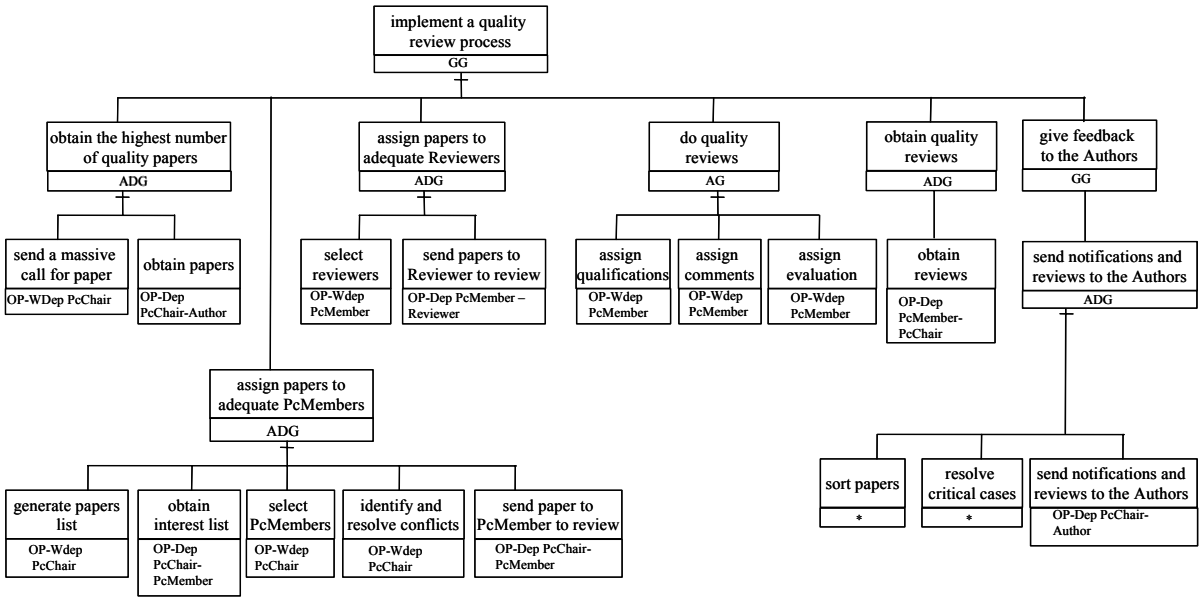

Figure 5 The GRT of the alternative *Implement a quality review process*

5.2 Step 2: Analyzing alternatives to satisfy goals using the GRT

Once the Goal-Refinement Tree is created, one of the alternatives to satisfy the organizational goal must be selected. The choice is based on trade-off analysis among the goals of the organizational actors. The strategy at this point is to decide the priority objectives to be satisfied in the business. For example, in the case study presented, if the goal *implement a simplified review process* is selected, the goals of the PcChair are satisfied, as there are papers to be presented and the review process is simple. However, this alternative does not satisfy the goal of the Authors. The Authors expect not only to have a paper accepted in the workshop, but also expect to obtain some (quality) feedback for their submission. Hence, the goal *implement a simplified review process* does not satisfy the Authors objectives. For this reason, we select the goal *implement a quality review process*.

The Goal-Refinement Tree can also be used to carry out obstacle analysis, conflict management and goal consolidation to generate a consistent and non-redundant goal structure. We propose using the methods and strategies proposed by KAOS [Dar93] and GBRAM [Ant96] to carry out these specific goal analyses.

As result of this phase, we have a Goal-Refinement Tree, which represents the objectives of the organization.

5.3 Step 3: Creation of the Strategic Models using the GRT

The objective of this phase is to use the knowledge elicited using the GRT to create the organizational models of the Tropos Framework.

5.3.1 Reasons to use the GRT to create an organizational model

In the current goal-based elicitation methods, the low-level goals are used to obtain the requirements of the information system. In this way, the operations that satisfy the goals are considered as software requirements. This approach is simple, cheap and consumes little effort. However, in this research works, the design decisions are taken too early, and the requirements are generated without the knowledge of the performance of the organization. In this approach, it is not possible to know if there are bottlenecks in the processes, or if there are organizational actors with a large number of tasks or dependencies. Therefore, it is not possible to improve the organization before generating the requirements of the information system.

We propose using the GRT to create an organizational model which allows us to perform this business analysis (business process reengineering analysis, dependency analysis, and task analysis*)* before taking decisions on the functionality of the information system. This allows us to have an improved organizational model that could be used to take design decisions

We propose to represent the organizational models using the Tropos Framework. The benefits of the approach a manifold:

a) perform reengineering analysis to obtain an improved organizational model
b) to map the organizational model in a specific organizational architectural style [Cas02] according to the goals of the organization [Bas03],
c) to insert the software system actor in the organizational model and redirect the actors responsibilities towards the software actor,
d) to map the organizational model into requirements of the information system.

5.3.2 Creation of the Strategic Dependency Model

The Goal-Refinement Tree is the starting point for the generation process of a Strategic Dependency Model. This model is focused on representing only the goals, tasks, resources and softgoals where there are dependencies between actors. Therefore, to create this model, it is necessary to take a subset of the Goal-Refinement Tree (the goals in which a dependency between actors exist). Note that in GRT there are goals, sub-goals or operational goals that could be satisfied by the actor itself. For example, the subgoal *do quality reviews* is performed only by the PcMember without dependencies with other actors. These kinds of goals are not represented in the Strategic Dependency Model.

We present the steps to achieve the translation of the GRT in the SD Model.

The first step is to use the organizational actors of the Goal-Refinement Tree to create the actors of the Strategic Dependency Model. The identified actors with responsibilities to satisfy goals or achieve operations in our case study are: PcChair, PcMember, Reviewer and Author.

The second step is to use the Achievement-Dependency Goals of the Goal-Refinement Tree to create the goal dependencies in the Strategic Dependency Model. As was mentioned in 5.1, the Achievement-Dependency Goals are goals that are refined Operational Goals where at least one of these is an Operation-Dependency. Therefore, these kinds of goals represent dependency relationships between actors.

To create a goal dependency, the actor responsible for achieving the Achievement-Dependency Goals must be identified. Later, the goal description must be analyzed to obtain the elements of a goal dependency in i*: the *depender*, *dependum* and *dependee*. In our case study, the Achievement-Dependency Goal *assign papers to adequate PcMembers* is translated into a goal dependency with the same name between the PcChair and the PcMember. In this dependency, the PcChair is the *dependee* actor because it executes the paper assignment operation. The PcMember is the *depender* actor because it depends on the assignment of the paper to be reviewed. Figure 6 shows the dependency generated for the goal *assign papers to adequate PcMembers*.

The third step is to use the Operation-Dependency of the Goal-Refinement Tree to create the resource and task dependencies of the Strategic Dependency Model. As was commented in 5.1 before, the Operation-Dependencies are goals that involve more than one actor for their execution. The Operational Goals performed by a single actor represent the internal actions of each actor in the Strategic Rationale Model.

An Operation-Dependency must be translated into a task dependency if the actor that depends on the execution of the operation specifies a particular way of doing it.

An Operation-Dependency must be translated into a resource dependency if the *depender* actor depends on the delivery of a resource to complete the operation.

There are cases where it is only necessary to generate a Resource Dependency; this is the case in which the *depender* actors do not specify the way of delivering the resource. There are cases where it is necessary only to generate a Task Dependency. This is the case in which the *depender* actor only needs to execute an action, but no resources are generated as result of this action.

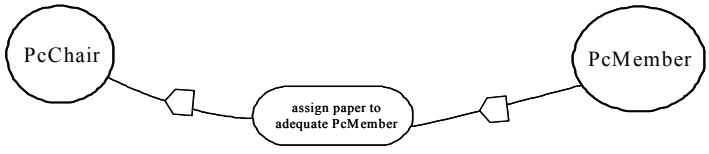

Figure 6 Dependency Goal *assign paper to adequate PcMembers*

In our case study, the Operation-Dependency *obtain reviews* is translated into two dependencies: a) the task dependency *Send Reviews* between the PcChair and the PcMember, and b) the resource dependency *Reviews* between the PcChair and the PcMember. The task dependency indicates that the PcChair determines the best way to send the reviews (for example, the PcChair may specify that the review must be sent using a special software). The resource dependency indicates that the PcMember depends on the PcChair to obtain the papers reviews. Similar tasks and resource dependencies must be created in the case of the dependency *Send Reviews* between the PcChair and the Reviewer. Figure 7 shows the dependency generated for the Operation-Dependency *obtain reviews*.

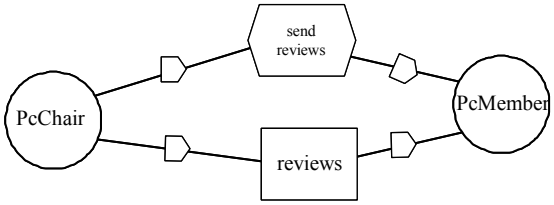

Figure 7 Dependencies generated for Operation-Dependency *obtain reviews*

Figure 8 shows the application of the translation process to the goal *implement a quality review process*. The next step of the process is to create a Strategic Rationale Model to detail the internal goal of the organizational actor

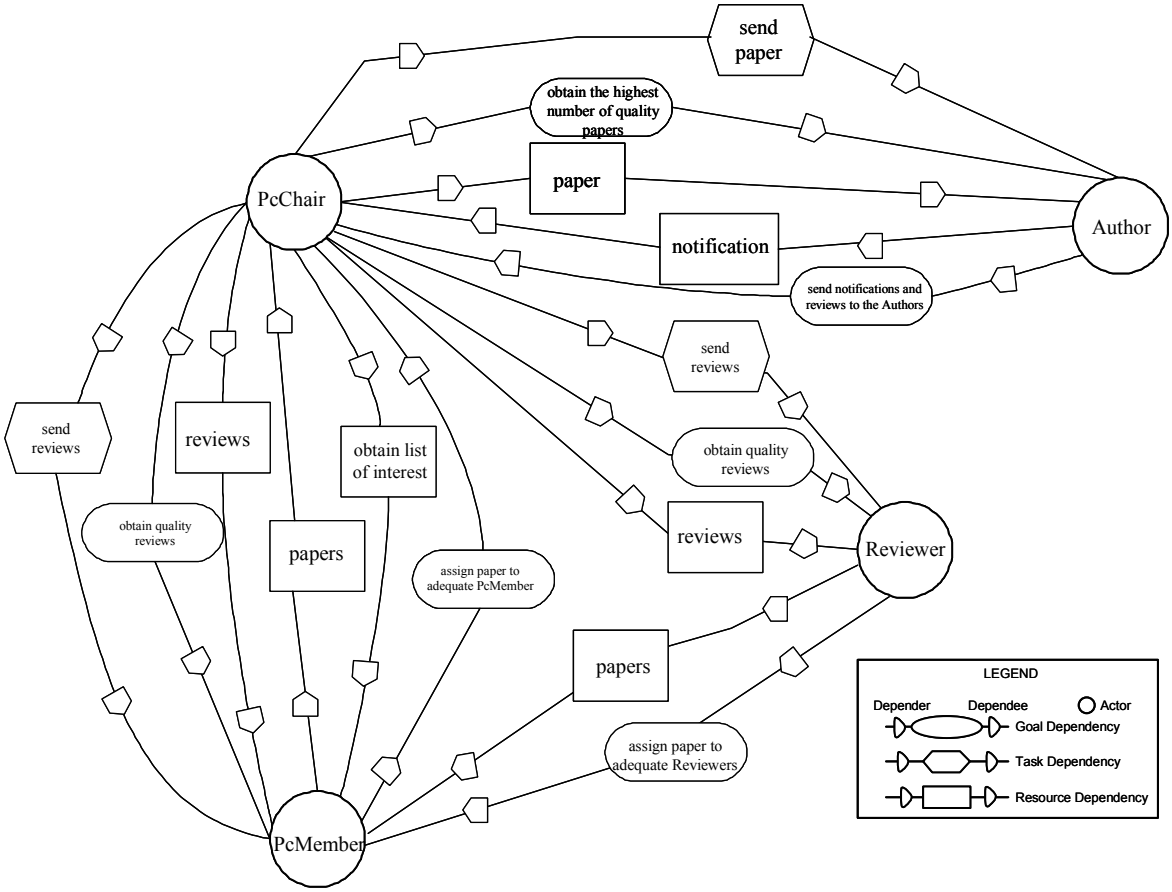

Figure 8 Strategic Dependency Model for the goal *implement a quality review process*

5.3.3 Creation of the Strategic Rationale Model

The objective of the Strategic Rationale Model (SR Model) is to define the internal tasks that the actors carry out in order to fulfill their dependencies with other actors. However, in the Goal-Refinement Tree, goals or actions may be performed by the actor without dependency relationships. In this sense, there may be a subset of the GRT in which neither the subgoal nor the operation implies dependencies between actors (For example, all the subgoals of the subgoal *do quality reviews* are goals without dependencies with other actors.). Therefore, with the GRT, SR Models can be created that include not only the actions refined into external dependencies, but also the actions refined into internal goals.

To create the SR Model, the Achievement Goals and the Achievement Dependency Goals of the Goal-Refinement Tree are translated into internal tasks in the Strategic Rationale Model. To do this, it is necessary to detect the actor responsible for satisfying the goals. Later, task decomposition links of the Tropos Framework are used to create internal task-refinement trees en each organizational actor of the SR Model. The Achievement Dependency Goals will be translated into internal tasks linked to the goal, resource or task dependencies between the actors. The Achievement Goals will be translated into internal task without dependencies with other actors.

In our case study, for example, the Achievement Dependency Goal *assign papers to adequate PcMember* is translated into the root of an internal task-refinement tree (with the same name as the goal) inside the actor PcChair. Later, the Operational Goals *generate paper list, obtain interest list, select PcMembers, identify and resolve conflict* and *send papers to PcMember to review* are translated into tasks and linked with the root goal of the internal tree. Figure 9 shows the task-decomposition generated for the ADG *assign papers to adequate PcMember.*

In the case of the Operation-Dependency of the Goal-Refinement Tree that has been derived in task dependencies in the Strategic Dependency Model, the execution of the task in the *dependee* actor must be indicated. To do this, an internal task in the *dependee* actor with the same name as the Operation-Dependency needs to be created and linked to the task dependency. In our case study, for example, the task dependency *send reviews* between the PcChair and the PcMember lead to the internal task *send reviews* in the actor PcMember. In the case of the Operation-Dependency of the GRT that has been derived in resource dependencies of the Strategic Dependency Model, the delivery of the resource in the *depender* actor must be identified. To do this, an internal task in the *depender* actor needs to be created to indicate the delivery of the resource and it must be linked to the resource dependency. In our case study, resource dependency *reviews* between the PcChair and the PcMember lead to the internal task to *obtain quality reviews* in the actor PcChair. Figure 10 shows the task generated for the goal *obtain reviews*

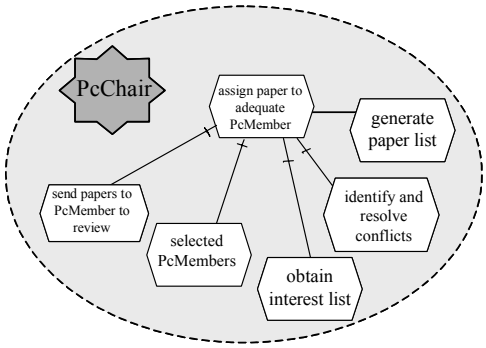

Figure 9 Task-decomposition generated for the ADG *assign papers to adequate PcMember.*

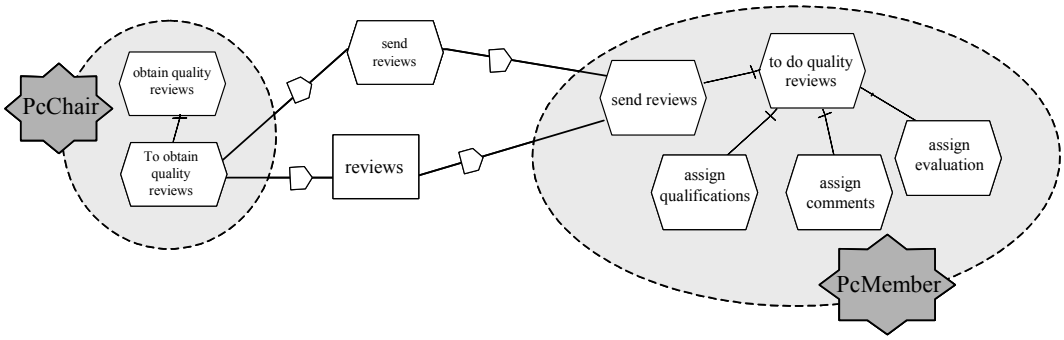

Figure 10 Tasks generated for the goal *Obtain Reviews*

Figure 11 shows the application of the translation process to the goal *implement a quality review process* case study analyzed.

5.4 Step 4: Including the software system actor in the organizational model.

Prior to generating the software requirements, we propose the inclusion of the software system actor in the organizational model. In doing so, the candidate operations to be automated are isolated.

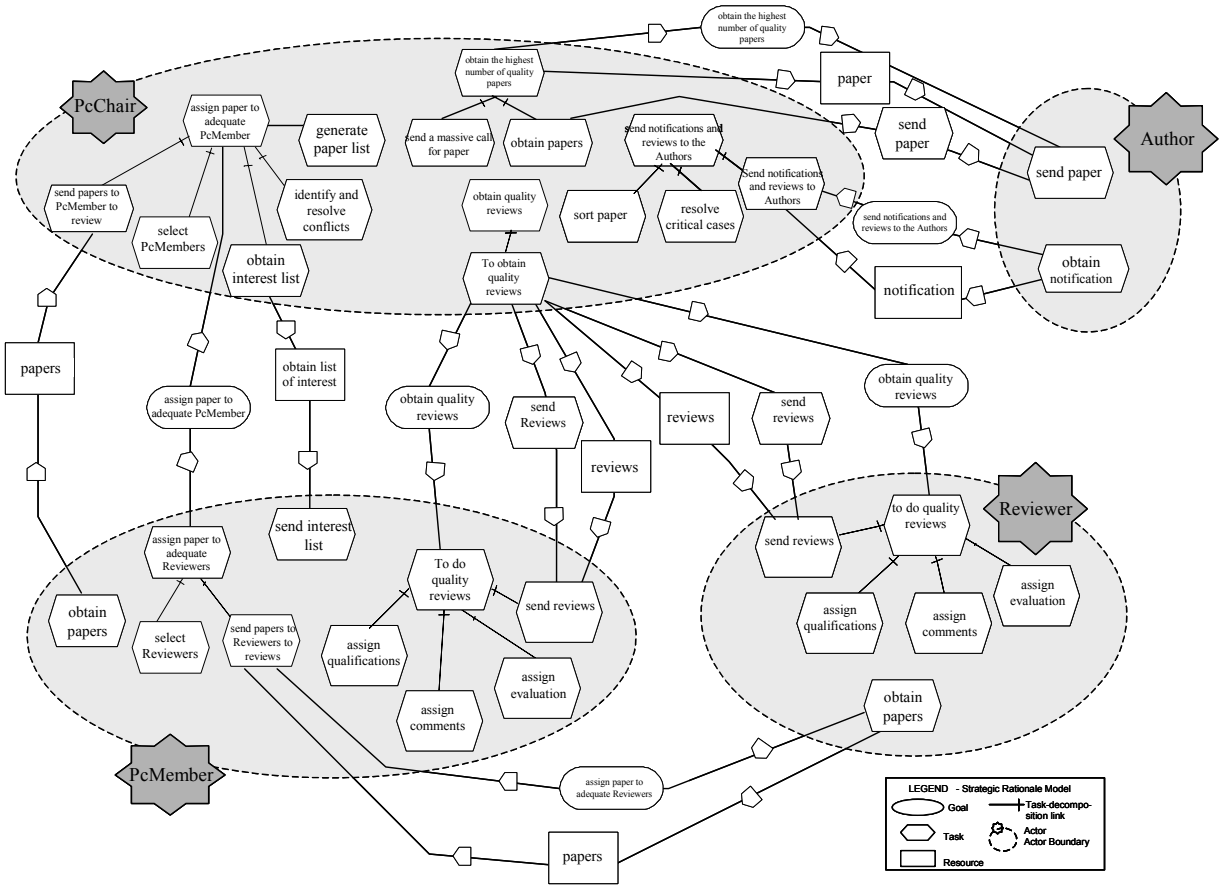

Figure 11 Strategic Rationale Model for the goal *implement a quality review process* case study

An important concept used in this process is the concept of "*module*". A module represents an internal task-refinement tree in an actor in the Strategic Rationale Model. An actor may have more than one module. This indicates that the actor should fulfill more than one goal in the organizational model. In our case study, the PcChair has the modules: *assign paper to adequate PcMembers, obtain the most quantity of quality papers, obtain quality reviews and send notifications and reviews to the Authors*. (Figure 11).

We present a brief version of the guidelines that permit the insertion of the actor system into the organizational model. Figure 12 shows the result of the application of these guidelines in our study case.

Guideline 1. Insert the actor system into the organizational model and to identify the modules that need to be delegated to the information system.

Guideline 2. Move these modules from the organizational actors to the system actor. This is only necessary when the main task (module root) needs to be automated. To move each module, it is necessary to apply two steps. The first step is to create a copy of the module root in the system actor. The second step is to create a task dependency (with the same name as the module) between the organizational actor and the system actor. This task dependency indicates that the software system actor is now responsible for complete the task. There may be manual operations in the modules, where the system can only be used to send or receive information. In these cases, it is necessary to leave these manual operations in the modules of the organizational actor.

Guideline 3. There are tasks that require information from the organizational actors when these tasks are transferred to modules in the system actor. In this case, it is necessary to create new resource dependencies between the system actor and the organizational actors.

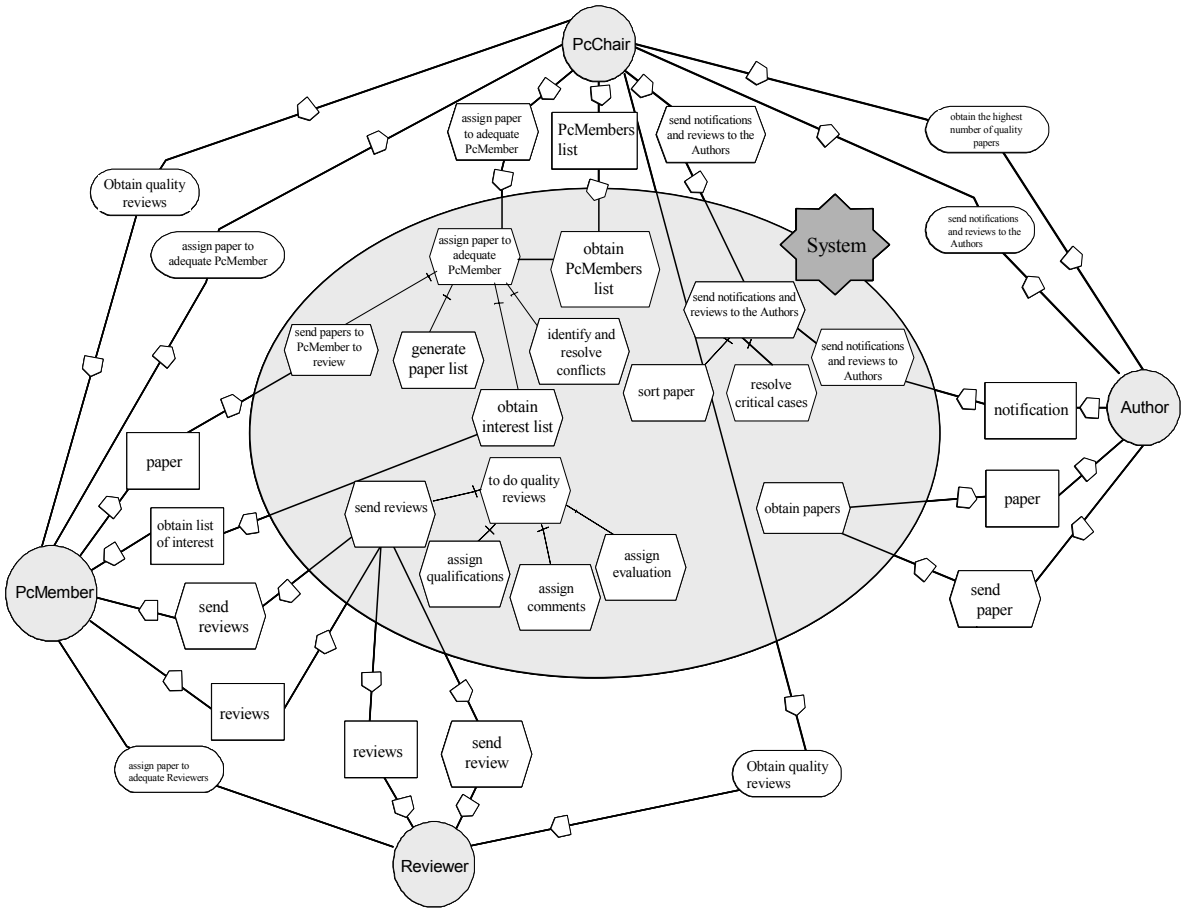

Figure 12 Insertion of the software system actor into the organizational model

This is the case with the task *generate PcMember list*, which requires information from the PcChair (when transferred to the system actor), as the system cannot generate the PcMember List by itself, since it requires the information from the PcChair.

Guideline 4. When resource dependencies are created, it is necessary to generate new tasks for sending and receiving resources inside the organizational actors.

The application of these guidelines allows organizational models to be semi-automatic translated into requirements models which are based on use case models and scenarios. These, in turn, can be translated into the user interface prototype [Mar02]. It is worth pointing out that this novel approach is quite effective.

6 Related Works

The most significant works in Goal-oriented requirements engineering are: a) KAOS [Dar93]: a formal framework based on temporal logic to elicit and represent the goals that the system software should achieve. b) GBRAM: a Goal- Based Requirements Analysis Method [Ant96] to represent the goals in an approach that is less formal but more focused on user needs.

In these works and in other goal-based approaches [Bub94][Bol02], the software requirements are obtained directly from the operations which satisfy the goal. The operations are mapped into use case model specifications or in services of the information system. This approach allows us to carry out the elicitation process at a level which is closer to the final users. However, this approach does not allow us to carry out analyses (business process reengineering analysis, dependency analysis, workflow analysis, task analysis) that are fundamental to obtaining requirements that reflect the functionality expected by the users of the information system.

In the organizational modeling field, the majority of the research efforts are focused on notations to specify the semantics of an organizational environment. However, a few efforts have been made to propose methodological support. In the UML community, the activity diagrams have been used for business modeling. Sequence diagrams can be used to represent the details of each activity diagram. However, the transition among these models is not straightforward.

7. Conclusions

A novel goal-based organizational modeling method has been presented in this paper. We define a set of steps to generate organizational models that reflect the goals of each actor as well as the general goals of the organization. We also describe the required steps to integrate the software system actor within the organizational model. That is, the software system actor will be assigned the responsibilities of executing goal and task. We have applied the guidelines for organizational modeling to a case study showing the graphic representation of each one of the organizational models generated.

The organizational models generated with the method presented in this paper could later be translated into a compatible UML use case specification, and its respective scenarios [San03]. The use case model generated serves as input to a semi automatic process that generates the specification of the system behavior as well as the prototype of the user interface [San01].

The method proposed in this paper allows us to create organizational models in a methodological process, by doing this; we go a step further in the process of properly embedding early requirements engineering into the software production process. In the actual research efforts in business modeling, the organizational models are still generated in ad-hoc fashion.

References

[Ant96] Anton Annie, “Goal Based Requirements Analysis”, in Proceedings Second International Conference on Requirements Engineering. ICRE ’96, pp. 136-144, April 1996.

[Bas03] Bastos Lucia, Castro Jaelson, “A Proposal for Integrating Organizational Requirements and Socio-Intentional Architectural Styles in Tropos”, STRAW03 - Second International Workshop From SofTware Requirements to Architectures, Portland, Oregon, USA. 2003.

[Bol02] Davide Bochini, Paolo Paolini, “Capturing Web Application Requirements through Goal-Oriented Analysis”, Proceedings of the Workshop on Requirements Engineering (WER 02), pp. 16-28, Valencia, Spain, 2002.

[Bub94] Bubenko, J. A., jr and M. Kirikova, “Worlds in Requirements Acquisition and Modelling”, in: Information Modelling and Knowledge Bases VI. H.Kangassalo et al. (Eds.), IOS Press, pp. 159– 174, Amsterdam, 1995.

[Cas02]Castro J. Kolp M. Mylopoulos J. “Towards Requirements-Driven Information Systems Engineering: The Tropos Project”. Information System Journal, Elsevier, Vol 27, pp 365-389, 2002.

[Ces02] Cesare S. Mark Lycett, “Business Modelling with UML, distilling directions for future research”, Proceedings of the Information Systems Analysis and Specification (ICEIS 2002), pp. 570-579, Ciudad-Real, Spain, 2002.

[Dar93] Dardenne, A. Van Lamsweerde and S. Fickas, “Goal Directed Requirements Acquisition,” Science of Computer Programming, vol. 20, pp. 3-50, North Holland, April 1993.

[Lou95] Loucopoulos Pericles, Evangelioa Kayakli, “Enterprise Modelling and the Teleological Approach to Requirements Engineering”, International Journal of Cooperative Information Systems (IJCIS), pp. 45-79 , 1995.

[Mar02] Martínez Alicia, Hugo Estrada, Juan Sánchez, “From Early Requirements to User Interface Prototyping: A methodological approach”, Proceedings 17th IEEE International Conference on Automated Software Engineering (ASE 2002), pp. 257-260. Edinburgh, UK. September 2002.

[Mcd96] McDermid, J..A, “Software Engineer´s Reference Book”, Butterworth-Heinenmann Editorial, 1996.

[San03] J. Sánchez; O. Pastor; H. Estrada; A. Martínez, “Semi Automatic Generation of User Interfaces Prototypes from Early Requirements Models”, Perspectives on Requirements Engineering. Editors: J. Leite; J. Doorn, Kluwer Academic Press, to appear in June 2003.

[San01] Sánchez J; Pastor O; Fons J.J. “From User Requirements to User Interfaces: a methodological approach”, Proceedings of the Fourteenth International Conference on Advanced Information Systems Engineering (CAISE 2001). pp. 60-75. Switzerland. 2001.

[Yu95] Yu, Eric, “Modelling Strategic Relationships for Process Reengineering”, PhD Thesis, University of Toronto, (1995).