Goal-based business modeling oriented towards late requirements generation[1]

Hugo Estrada[1,2], Alicia Martínez[1,3], Oscar Pastor[1]

[1] Technical University of Valencia
Avenida de los Naranjos s/n, Valencia, Spain
{hestrada, alimartin, opastor}@dsic.upv.es
[2]CENIDET Cuernavaca, Mor. Mexico
[3] I.T. Zacatepec, Morelos, Mexico

Abstract. Recently, a lot of research efforts in software engineering have focused on integrating business 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 to generate business models and to use these to generate a requirements model. This lack of both generation methods and traceability relationships between models makes practical application in software development enterprises difficult. The objective of this paper is to define a goal-based methodological approach for the generation of business models and to use these models as the starting point for the process of software requirements specification. This will enable us to develop information systems that integrate the necessary functionality so that the business actors perform their tasks and fulfill their goals.

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 [Mcd94] 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 of a business process, allowing the business actors

[1] This project is partially funded by the Asociación Nacional de Universidades e Instituciones de Educación Superior ANUIES, Mexico

David Juárez Romero, J. Crispín Zavala-Díaz,(eds.). AGECOMP2004, Memorias del 3er Congreso de Cómputo AGECOMP, UAEM, México, ISBN(e) 968-878-205-X. 80 - 93

to reach their individual goals, as well as the general goals of the organization. In this context, there are research works that highlight the importance of using business models as a starting point in the development of information systems [Bub94][Ces02][Lou95][Cas02]. Unfortunately, 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 to generate business models and use these to generate a requirements model.

In this paper, we present a novel method to create business models which are represented in the i* Framework. Specifications of information system requirements (use case and scenarios) are also derived from these business models. Hence, two important and complementary approaches are discussed in this paper. First, we begin by defining a goal-oriented method to be used to construct the business models. Later, we discuss an approach to generate a requirements model (represented through the uses cases and their corresponding scenarios) from the business model. This requirement specification is later used for the semi automatic creation of the user interface [San03].

The paper is structured as follows: Section 2 presents the background of the proposed method. Section 3 describes an overview of the proposal. Section 4 presents the Goal-based elicitation Method. Section 5 presents the construction of the i* Business Models. Section 6 presents the elicitation process of system requirements based on the business models. Finally, Section 7 presents the conclusions.

2 Background

This section presents the main concepts that are used in this paper: goal modeling and business modeling. Both approaches are combined to create a capture method of business requirements. The advantages of our proposal over other proposals are presented in this section.

2.1 Goal Modeling

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 operational goals that satisfy the goals. The operational goals are mapped into use case model specifications or into 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 business analysis (business process reengineering analysis, dependency analysis, workflow analysis, task analysis), which are fundamental to obtaining requirements that reflect the functionality expected by the users of the information system.

2.2 Business Modeling

Business Modeling is a set of techniques used to represent and to structure the knowledge of a business enterprise [Bub94]. The enterprise analysis allows us to determine with great precision: the operations that satisfy each one of the goals, the network of dependencies among actors, the sequence in which the tasks of each business process should be executed, the dependency type, the task to be automated, etc. This information is fundamental for the generation of a requirements model that gives real support to the business tasks.

There is a lot of research being done in this field; however, the i* framework [Yu95] is one of the most well-known techniques today. The i* framework allows us to describe business models made up of social actors that have freedom of action, but that depend on other actors to achieve their objectives and goals. This information is useful for viewing the tasks that each actor is expected to perform, as well as for analyzing the repercussions of the fulfillment or non-fulfillment of the tasks assigned to the actors. The i* framework has a graphical notation with few elements, which allows us to represent the actors, dependencies, resources and the tasks of the business process in a unified view. The majority of business modeling techniques use multiple diagrams (each diagram represents a specific view of the business) to have the same expressivity as the i* framework. The i* notation also allows us to include software systems as actors inside the business model. These characteristics distinguish the i* framework from the rest of the business modeling techniques [Bub94], [Ces02], [Lou95].

The i*framework is made up of two business models that complement each other: the Strategic Dependency Model and the Strategic Rationale Model.

The *Strategic Dependency Model* (SD) shows the dependencies that exist between business actors to achieve their goals, carry out tasks and provide or request resources. 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.

The SD model is represented by a graph where nodes represent actors (agents), and where the links represent dependencies. The combination of nodes and links in the SD Model create a network of dependencies that helps to graphically represent the external relationships between actors. There are directed arrows to link the *depender*, *dependum* and *dependee*. These arrows indicate the direction of the dependency, determining which actor is the *depender* and which is the *dependee*.

The SD model is composed by four types of dependencies: a) 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. b) Resource dependency in which an actor depends on another actor to deliver a resource that can be either material or informational. c) Task dependency in which there exists a dependency for the carrying out of a task, establishing the way in which it should be performed. d) SoftGoal dependency. This is similar to the goal dependency, with the difference that the goal has not been precisely defined.

The Softgoal dependency, which corresponds to non-functional requirements, does not appear in our approach, because this paper is focused on generating Use Cases, that is a functional description of the information system.

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 options that can be taken to fulfill a task or goal.

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

3. Description of the Proposal

The objective of the proposed method "Goal-based Business Modeling oriented towards late requirements generation" is to help to construct the software requirements specification using a business model as the starting point.

The steps of our proposed method are summarized as follows:

1. Use a Goal-Based Elicitation Method to construct a Goal-Refinement Tree (GRT) which captures the organizational context.
2. Use the GRT to create the Strategic Models of the i* Framework. These Models could then be used to perform business improvement analysis.
3. Use the Strategic Models to derive functional (use case) specifications with their corresponding scenarios [San03].

In order to illustrate our approach, we used the *Conference Review Process* case study. The purpose is to model the business process to obtain a software system that handles the process of submission, assignation, evaluation and selection of papers for a conference.

4. Goal-based Elicitation Method

The Goal-Based Elicitation Method proposed in this paper allows us to elicit the business 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 (GRT) 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, all the leaves represent operational goals that satisfy the low-level goals.

4.1 Goal Classification

We propose a goal classification to structure the goals in the GRT. The goal classification was created to represent not only the internal goals or operations of the business actor, but also to represent the cases where there are relationships among actors. Relations of this kind imply that the actors depend on other actors to satisfy their goals or perform their operations. These relations are fundamental for creating the Strategic Models of the i* Framework. For this reason, the goal classification is not exhaustive; we classify only the goals necessary to create a i* business model.

Operational Goals: They are performed by the correct state transition of one of the business actors and change the state of one or more objects [Dar93]. They are characterized by pre-, post- and trigger- conditions. 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. This kind of Operational Goal is 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. This kind of Operational Goal is represented in the GRT as OP-WDep.

- **Achievement Goals**: These goals are refined in Operations Without-Dependency or in other Achievement Goals. They are represented in the GRT as *AG*.
- **Achievement-Dependency Goals**: These goals are refined in Operational Goals, where at least one of these is an Operations-Dependency or in another Achievement-Dependency Goal. 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 classes of relationships between goals to structure the Goal-Refinement Tree.

- **Conflict Goals**: This is the case where the achievement of a goal has a negative impact on the satisfaction of another goal or subgoal. They are represented by CG.
- **Decomposition links**: 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 links**: This is the case in which only one of the different alternatives to satisfy a goal could be satisfied. These represent a decision structure to show the alternatives that exist to achieve a goal. These goals are represented using the link ▶ which links the alternative subgoals with the more general goal.

In this paper, we have determined that all the selected goals to model should be derived in operations (Operation-Dependency or Operation Without-Dependency), in the same way that all operations should be connected to goals. This restriction prevents the specification of goals that are not reflected in operations, as well as the specification of operations that do not satisfy any goal. The operationalization of the goals is one of the fundamental steps for construction in a goal model.

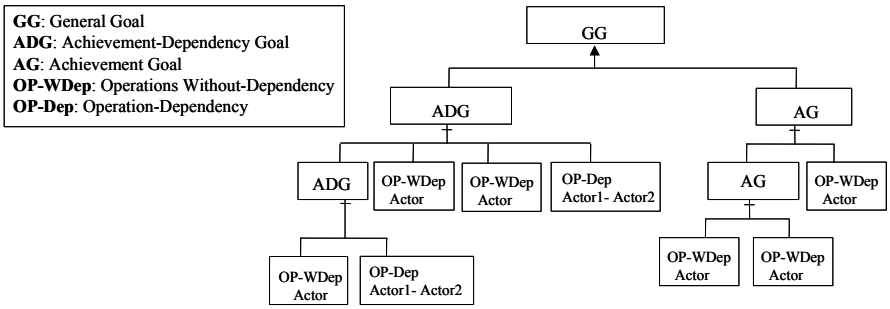

Fig. 1. Example of a hierarchical structure of the Goal-Refinement Tree

Figure 1 shows an example of a hierarchical configuration in the GRT.

4.2 Goal Elicitation

To create the Goal-Refinement Tree, it is possible to use a strategy by refinement and/or by abstraction.

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 them. 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 that are responsible for achieving them.

In the *abstraction strategy*, it is necessary to detect the actors that participate in the organization. Once the business 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.

In the process of identifying the actors responsible for achieving the Operational Goal, there may be dependency relationships among actors. There are dependency relationships when another actor is needed to provide a resource or perform an operation. These dependencies must be represented in the Goal-Refinement Tree as Operation-Dependency.

To illustrate our case study, we have shown the textual description of the Goal-Refinement Tree (Table 1). The first column shows the identifying goals. The second column indicates the goal type. The third column shows the actors involved in reaching that goal or operation. If there are two actors in the third column, this indicate a relationship between actors.

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

Name of the goal	Goal type	Actors
Goal: to perform a paper review process	*GG*	
Obtain the highest number of quality papers	*ADG*	PcChair - Author
• Send a call for papers	*OP-Wdep*	PcChair
• Obtain papers	*OP-Dep*	PcChair - Author
Assign papers to adequate PcMembers	*ADG*	PcMember - PcChair
• generate paper list	*OP-Wdep*	PcChair
• obtain interest list	*OP-Dep*	PcChair - PcMember
• select PcMembers	*OP-WDep*	PcChair
• identify and resolve conflicts	*OP-WDep*	PcChair
• send papers to PcMembers to review	*OP-Dep*	PcMember - PcChair
Assign papers to adequate Reviewers	*ADG*	PcMember - Reviewer
• select Reviewers	*OP-Wdep*	PcMember
• send papers to Reviewers to review	*OP-Dep*	PcMember - Reviewer
To do quality reviews	*AG*	PcMember
• assign qualifications	*OP-WDep*	PcMember
• assign commnents	*OP-WDep*	PcMember
• assign evaluation	*OP-WDep*	PcMember
Obtain quality reviews	*ADG*	PcChair - PcMember/ Reviewer
• obtain reviews	*OP*	PcChair - PcMember/ Reviewer
Give feedback to the Authors	*GG*	
• Send notifications to the Authors on time	*ADG*	PcChair - Author
• sort papers	*	
• resolve critical cases	*	
• send notifications and reviews	*OP-Dep*	PcChair - Author

Table 1. Goal-Refinement Tree for the Conference Review Process case study.

5. Using the GRT to create the Strategic Models of i* Framework

The objective of this phase is to use the knowledge elicited using the GRT to create the business models of the i* Framework.

In the current goal-based elicitation methods, the low-level goals are used to obtain the requirements of the information system. However, in this approach, the design decisions are taken too early, and the requirements are generated without the knowledge of the performance of the organization. This approach focuses on generating software specifications rather than on supporting reasoning and analysis about the performance of the business process.

Using only a goal-based structure, it is not possible to show: the order of execution of the operations, the work product flow, the workflow, the summarization of responsibilities of each business actor, etc. Therefore, it is not possible to improve the organization before generating the requirements of the information system.

We propose using the GRT to create a business 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 business model that could be used to take design decisions.

This paper is focused on generating functional specifications of the system software. For this reason, the Softgoal Dependencies of the i* Framework, related to non-functional requirements are not consider in this paper.

5.1 Creation of the Strategic Dependency Model

The Goal-Refinement Tree is the starting point for the generation process of a business Model represented in the i* framework. The process begins with the creation of a Strategic Dependency Model (SD Model). The SD Model is focused on representing the dependency relationships that exist among the organizational actors. For this reason, this model must be constructed using a subset of the GRT (the goals in which a dependency exist between the actors).

The first step is to use the organizational actors of the GRT to create the actors of the SD Model. The actors identified 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 GRT to create the goal dependencies in the Strategic Dependency Model. As was mentioned in 4.1, 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. In our case study, for example, 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.

The third step is to use the Operation-Dependency of the GRT to create the resource and task dependencies of the Strategic Dependency Model. As was mentioned in 4.1, 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. For example, in the GRT of the 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. Figure 2 shows the result of the application of the translation process to the case study analyzed.

The SD Model is useful for detecting potential problems with the performance of the business model for finding: actors with a large number of dependencies, actors that represents bottlenecks, redundant dependency relationships, etc. This information can be used to improve the business model.

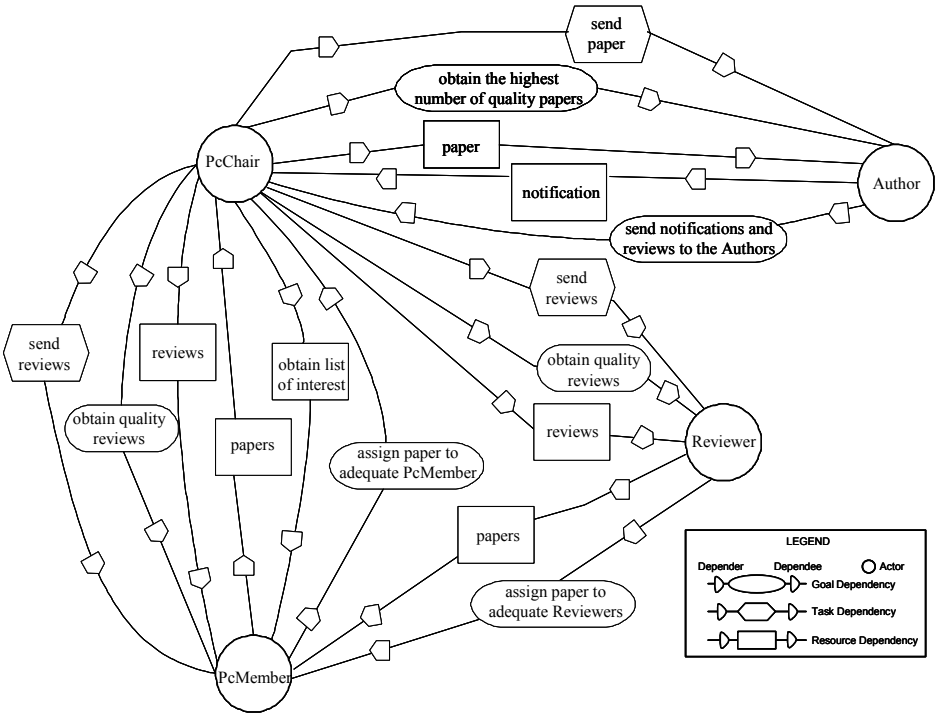

Fig. 2. Strategic Dependency Model of the Conference Review Process case study

Once the SD Model is created, the Strategic Rationale Model must be created in order to detail the internal tasks that accomplish the dependencies.

5.2 Creation of the Strategic Rationale Model

The construction of the Strategic Rationale Model (SR Model) consists in defining the internal operations that all actors carry out in order to reach their dependencies. To do this, the Achievement Goals of the Goal-Refinement Tree are translated into internal goals or internal tasks in the Strategic Rationale Model. This is done using task decomposition to create internal task-refinement trees in each business actor. Some of these internal goals or tasks will be connected with the task dependencies or resource dependencies defined in the Strategic Dependency Model.

In our case study, for example (Figure 3), 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.

In the case of operations of the GRT that have been derived in resource dependencies, it is necessary to indicate the delivery of the resource in the *depender* actor. To do this, an internal task must be created in the *depender* actor to indicate the

delivery of the resource and link it to the resource dependency. In our case study, resource dependency *Reviews* between the PcChair and the PcMember lead to the internal task *Obtain Reviews* in the actor PcChair. Figure 3 shows the SR Model generated. The internal goals of each actor are represented in this model.

6. Using the business models for software requirements elicitation

The objective of this phase is to rely on the business model as the starting point of the process of requirements specification. The initial step in this process is to include a software system actor in the business model. In doing so, the candidate operations to be automated are isolated.

6.1 Process of insertion of the system actor

The strategy of this process consists of determining the type of interaction of each business actor model with the software system actor. An important concept used in this process is the "module". A module represents an internal task-refinement tree in an actor in the Strategic Rationale Model.

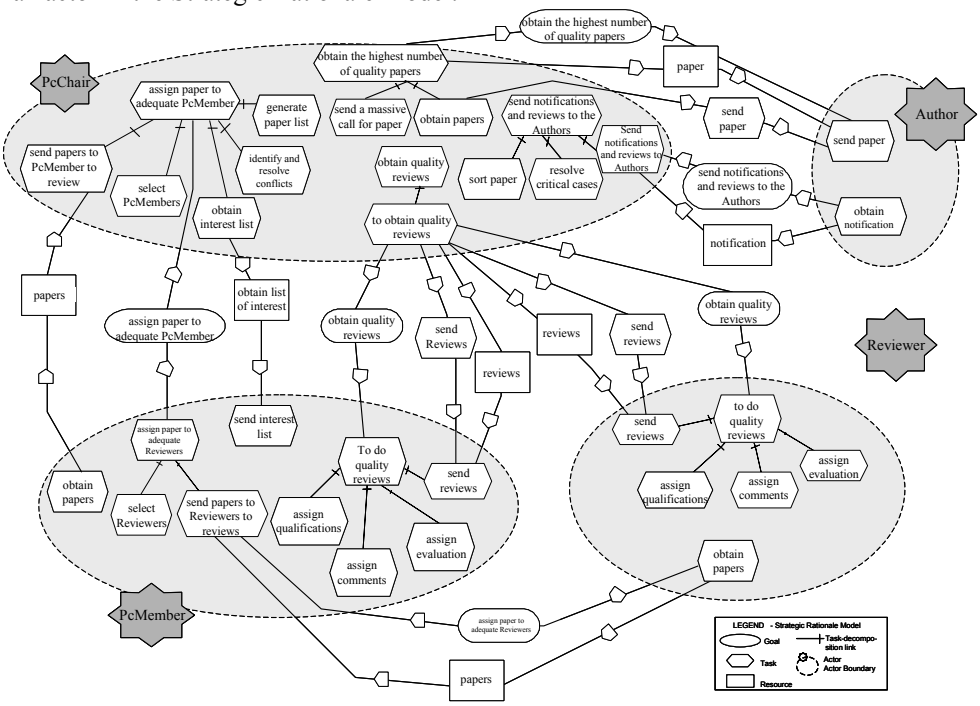

Fig. 3. SR Model of the Conference Review Process case study without system actor

An actor may have more than one module. This indicates that the actor should fulfill more than one goal in the business model.

In our case study, for example, the PcChair has the modules *Send Notification* and *Assign Papers* (Figure 3).

We present a brief version of the guidelines that permit the insertion of the actor system into the business model. Figure 4 shows the result of the application of these guidelines to our case study.

Guideline 1. Insert the actor system into the organizational model and 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 completing 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. 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.

In the SR Model generated (Figure 4), the tasks selected to be automated have been redirected towards the system actor. In this way, the system actor has the following functions to help the users to accomplish the business task: *assign papers to adequate PcMembers*, s*end notifications and reviews to Authors obtain papers*, and *to do quality reviews*. Each one of these functions is refined in a subtask in the SR Model.

The application of these guidelines allows business models to be naturally translated into requirements models which are based on use case models.

6.2 Use case Generation from a business model

Santander and Castro [Sant01] have previously studied the generation of use case models from business models. This approach focuses only on the translation process of the business models in a use case model specification. Therefore, this work does not focus on the problem of business model generation. We place more emphasis on business model creation by providing guidelines that allow us to generate business models adapted for use case generation.

To guide the process of mapping between the business models and the use case model specified in UML, we defined a set of steps to establish the correspondence between the elements of the business model specified in the i* framework (with the system actor integrated in explicit form) and the use case model and the corresponding scenarios.

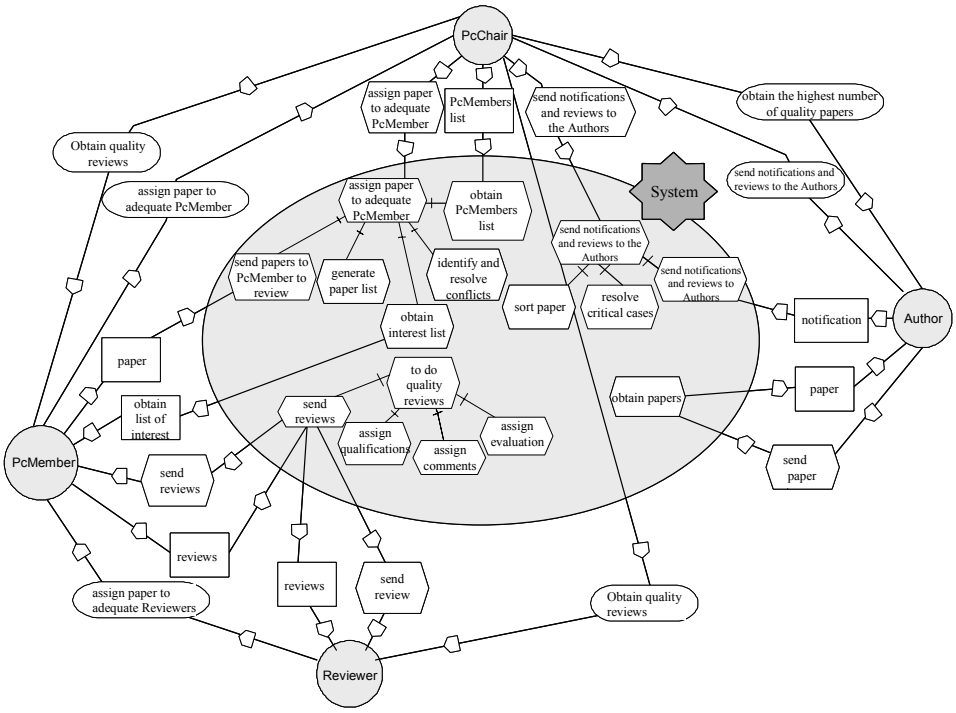

Fig. 4. Insertion of the software system actor into the business model

Obtaining a use case model

The first step of the process is to determine the relevant goal dependencies that will be used to generate the use case model. The relevant dependencies are those goal dependencies which were translated into resource or task dependencies between a business actor and the system actor. If a goal dependency has not given rise to dependencies with the system actor, then it represents a manual activity in which there is no interaction between the user and the software system. For this reason, this goal does not need to be modeled as a use case.

Obtaining the use case actors

The second step of the process is to determine the use case actors. In order to do this, it is necessary to analyze those resource and task dependencies which are derived from the relevant goal dependencies. The software system actor is always present at one end of the resource or task dependency, and the business actor is always present at the other end. Therefore, the business actor appears as a system user and also as a use case actor. The result of the application of these steps to our case study is shown in Figure 5.

Scenario representation

It is necessary to create a description model for each goal dependency which has been chosen as a use case. Templates are used to show a sequence of events between the actors and the system. To do this, we use a variant of the template proposed by L. Constantine [Con99](for details, consult Sánchez [San01]).

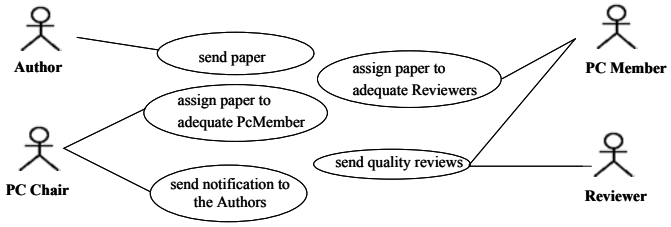

Fig. 5. Initial Model of the Conference Review System

It is necessary to detail the user intentions as well as the system responsibilities in the template. This division allows us to detect when the actors request services and when the system acts as a supplier of information. This information is obtained from the Strategic Rationale Model using the steps shown in [Mar02]. To illustrate the template, we select the use case *Send Reviews* shown in Table 2.

Use Case Name: Send Review	
Include:	None
Extend:	None
Preconditions	The PCMember has logged into the system.
Post conditions	The system saves a new review information
Primary Actor	PcMember
Secondary Actors	None
Roles	Reviewer
User intentions	**System responsibilities**
1. The PCMember selects "Enter Review"	2. The system asks for Paper ID.
3. The PCMember introduces Paper ID	4. The system verifies paper ID and the system displays paper title.
	5. The system asks for review data
6. The PCMember introduces values for Originality, Technical Quality, Relevance and Overall Rating.	7. The system asks for reviewer's comments
8. The PCMember introduces Author and Program Committee comments, and selects "Apply".	9. The system saves the review information.
Asynchronous extensions	
The Reviewer can select *Stop* at any point	
Synchronous extensions	
If there is no paper ID, the system displays an error message at point 4.	

Table 2. Specification of the use case *Send Review*

7. Conclusions

A goal-based business modeling method has been presented in this paper. We define a set of steps to generate business models that reflects the goals of each actor, as well as the general goals of the business. We also present the required steps to integrate the software system actor inside the business model. The software system actor integrates the business task to be automated by the information system. We have

applied the guidelines for the business modeling to a case study, showing the graphic representation of each one of the business models generated. Finally, the steps to translate the business model into a compatible UML use case specification and its respective scenarios are also presented. 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 [Mar02]. This specification is used in the last phase of the method to simulate the interfaces generated. The simulation process is supported by a tool which is programmed in Delphi, and which has a relational repository in Interbase. The tool generates prototypes in the Delphi, Java and HTML languages. This tool is being reprogrammed in Java language.

References

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

[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, Evangelia Kavakli, "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.

[Mcd94] McDermid, J..A, "Software Engineer´s Reference Book", Edit. Butterworth-Heinenmann, 1994.

[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.

[Sant01] Santander, V. F. A., Castro, J. B., "Developing Use Cases from Organizational Modeling", IV Workshop on Requirements Engineering (WER), Buenos Aires Argentina, 2001.

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