韓國經營科學會誌
第28卷 第2號
2003年 6月

사용 사례 모델링을 위한 시나리오 기반의 목적 지향 접근법에 관한 연구*

김정욱** · 박수용*** · 남기찬****

Scenario-based Goal-oriented Approach for Use Case Modeling*

Jeong-Wook Kim** · Sooyong Park*** · Ki-Chan Nam****

■ Abstract ■

Use case modeling in UML (Unified Modeling Language) is getting widely used in OOAD and CBD and it is considered a useful technique in dealing with the complexity of the requirements analysis. However, some of the problems with use case modeling are that it is not structured, difficult to handle non-functional requirements, and to analyze impacts among use cases. To alleviate these problems, we propose a scenario-based goal-oriented approach for use case modeling. The proposed approach is to apply a goal-oriented analysis method to use case modeling. Since goal-oriented analysis method is not systematic and many heuristics are involved, we have adopted scenarios as the basis for goal extraction. The proposed method is applied to CBIS (City Bus Information Subsystem) in ITS (Intelligent Transportation System) domain. The proposed approach helps software engineers to analyze the impact among use cases and represent non-functional requirements.

Keyword : Use Case, Requirement Engineering, Scenario-based Analysis, Goal

논문접수일 : 2003년 1월 25일 논문게재확정일 : 2003년 5월 28일
* 이 논문은 2002년 학술진흥재단 협동연구과제(KRF-2002-042-B00054)의 학술연구비에 의하여 지원되었음.
** 세종대학교 경영학과 부교수
*** 서강대학교 컴퓨터학과 부교수
**** 서강대학교 경영학과 부교수

1. Introduction

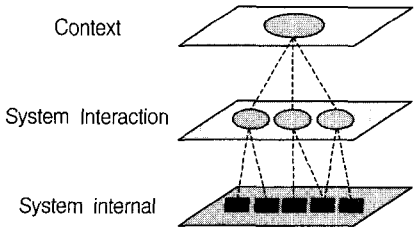
Use case driven analysis is getting popular in OOAD(Object-Oriented Analysis and Design) and CBD(Component-Based Development) for requirement elicitation, analysis and specification. It supports identifying and independently analyzing different use cases, which is a narrow aspect of the system usage at a time [1, 2]. But use case driven analysis is not an appropriate method to structure requirements and to analyze impacts among requirements. It doesn't provide rationale to justify software requirements and to represent non-functional requirements. In addition, novices have difficulty modeling the problem at hand at the appropriate level of abstraction for the use case. To resolve these limits, goal-oriented requirements analysis is proposed [4]. There have been some approaches in goal-oriented requirement analysis [6], but they lack a systematic process. Hence, we propose a scenario-based goal-oriented analysis method for use case modeling. By applying a goal-based approach to use case modeling, it is easy to structure the requirements. This can greatly assist in analyzing impacts, functional and non-functional requirements. It helps software engineers to prevent possible conflicts in late phases of the software development.

The rest of the paper is organized as follows. Section 2 presents the scenario-based goal-oriented analysis method. Section 3 presents the approach for applying the scenario-based goal-oriented analysis method to use case model. The proposed method is applied to a practical system development, namely, the City Bus Information system of ITS (Intelligent Transportation System) in Section 4. In section 5, the research related to goals and scenarios is presented. Finally, we discuss the essential properties of our approach, and future prospects for our research.

2. Scenario-based goal-oriented analysis method

2.1 Goal

We define a goal as objective to be achieved by the software system, where objectives correspond to requirements. From the definition, we divide goals into three levels of abstraction including context goal, system interaction goal and system internal goal. The three levels of abstraction helps developer to identify goals from problem domain and to refine goal. [Figure 1] shows goal's abstract level.



[Figure 1] The goal's abstract level

- Context Goal

Context goal is to identify the services that a system should ultimately provide to an organization and their rationale. Context goal is one way that a business goal is achieved.

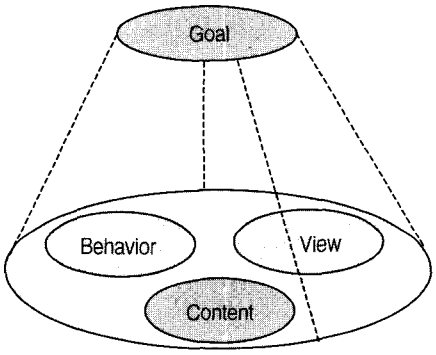
- System interaction Goal

System interaction goal focuses on the interactions between the system and its users. These interactions are required to achieve the services assigned to the system by the context goal.

• System internal Goal

System internal goal focuses on what the system needs to perform, i.e., the interactions needed at the system interaction goal level. This goal expresses a possible action and state to perform an interaction identified by the system interaction goal.

We classify goal into three aspects-Behavior, View, and Content. These three aspects of a goal are shown in [Figure 2].



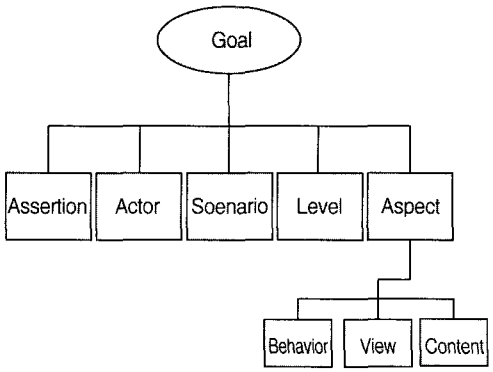
[Figure 2] The goal's three aspects

- Behavior aspect classification : Achieve, Maintain
- View aspect classification : Actor-specific, System-specific
- Content aspect classification : Functional, Nonfunctional

2.2 Goal specification

Based on our definition, abstraction level and aspect classification goal can be specified. The goal that is elicited in scenario-based goal-oriented analysis has several internal constituents. These internal constituents are necessary elements to correctly specify the goal. Goal specification is used in project plan, design, coding and testing after the requirement analysis phase.

[Figure 3] shows the internal constituents to specify the goal and briefly explained below.



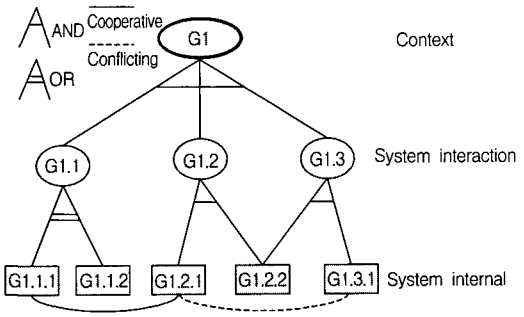
[Figure 3] Internal constituents to specify a goal

- Assertion : Short sentence that represents the goal is written.
- Actor : Actors associated with the goal are written
- Scenario : scenarios show sequences and states of input and outputs to describe the operation of the system. The scenario is written.
- Level : The goal's abstract level is written
- Aspect : The goal's classification by the goal's three aspects is written.

We define the specification form of the requirements being analyzed by scenario-based goal-oriented analysis method. The entire structure of the requirements being analyzed is represented as a goal hierarchy diagram. Each goal is specified in the proposed specification form.

2.3 Goal relationships

We analyze relationships among goals by adopting goal hierarchy diagram as shown in [Figure 4].

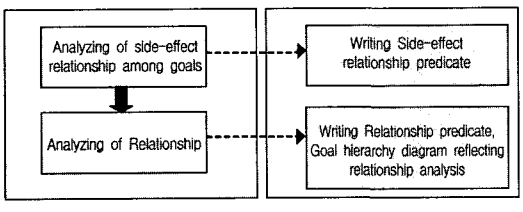


[Figure 4] The goal hierarchy diagram

AND & OR relations could exist between sibling goals, and they could be defined as follows :

- AND : These relationships among goals link together those goals that require each other to define a completely functioning system.
- OR : These relationships represent alternative ways of fulfilling the same goal.

Impacts relationships are analyzed among goals with respect to the goal's achievement. One goal may cooperatively or conflictingly affect another goal with respect to the upper goal's achievement. These impacts relationships help software engineers to resolve conflicts among software modules that may happen in design phase. Analysis of impacts relationships process is composed of two phases. [Figure 5] shows impacts relationships analysis process and its products.



[Figure 5] Impacts relationships analysis process and products

Impacts are analyzed among subordinate goals with respect to its upper goal's achievement. It is expressed by the predicates in <Table 1>.

〈Table 1〉 Relationship predicates

• $CP_{G1.1}(G1.1.1, G1.2.1)$ The goals G1.1.1 and G1.2.1 are cooperative with respect to goal G1.1's achievement.
• $CF_{G1.2}(G1.2.1, G1.3.1)$ The goals G1.2.1 and G1.3.1 are conflicting with respect to goal G1.2's achievement.

With respect to those upper goal's achievement, effects between subordinate goals are determined whether to be cooperative or conflicting. This analysis offers the basis for establishing impacts relationships between goals by the analyst. <Table 2> expresses the predicates of cooperative and conflicting side-effect relationship based on the predicates in <Table 1>.

〈Table 2〉 The Side-effect relationship predicates

• $CP(G1.1.1, G1.2.1) = CP_{G1.1}(G1.1.1, G1.2.1)$ $CP_{G1.2}(G1.1.1, G1.2.1)$ The relationship between the goals G1.11 and G1.2.1 is cooperative side-effect relationship.
• $CP(G.1.2.1, G1.3.1) = CF_{G1.2}(G1.2.1, G1.3.1)$ $CF_{G1.3}(G1.2.1, G1.3.1)$ The relationship between the goals G1.2.1 and G1.3.1 is conflicting side-effect relationship.

The relationship between the goals G_i and G_j is denoted as $R(G_i, G_j)$, and is defined as a pair of predicates $< CP(G_i, G_j), CF(G_i, G_j) >$, where $CP(G_i, G_j)$ is true if G_i is cooperative with G_j and $CF(G_i, G_j)$ is true if G_i is conflicting with G_j. There are four possible relationships between goals in <Table 3>.

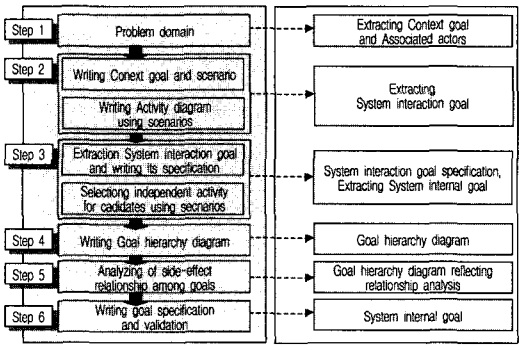
〈Table 3〉 Four possible relationships between goals

- $R(G_i, G_j)$ = <*False, False*>
  The G_i and G_j are irrelevant
- $R(G_i, G_j)$ = <*True, False*>
  The G_i and G_j are cooperative
- $R(G_i, G_j)$ = <*False, True*>
  The G_i and G_j are conflicting
- $R(G_i, G_j)$ = <*True, True*>
  The G_i and G_i are counterbalanced

Requirements with cooperative relationship offer rationale for designing components in the design phase. If requirements with conflicting relationship, which implies inconsistency among requirements, are resolved in the analysis phase, development cost is reduced.

2.4 Scenario-based goal-oriented analysis process

Scenario-based goal-oriented analysis acquires well-structured requirements by abstracting, characterizing, classifying, and refining the requirements with scenario-based goals. [Figure 6] shows the process of scenario-based goal-oriented analysis and it's products.

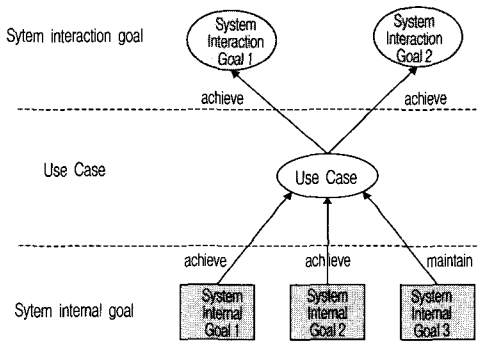


[Figure 6] Scenario-based goal-oriented analysi's process and products

3. Scenario based goal-oriented approach for use case modeling

3.1 Relationship between goal and use case

To apply goal-oriented method to use case model, examination of the conceptual relationship between the goal and the use case is needed. [Figure 7] shows the relationship diagram between goal and use case.



[Figure 7] The relationship diagram between goals and use case

Through goal-oriented analysis, we can create a goal-hierarchy diagram and the corresponding goal-specification. We can also create the use case model after use case-driven analysis. With these two models, we can generate diagrams to represent the relationships between use cases and goals.

- **System interaction goal and Use Case**

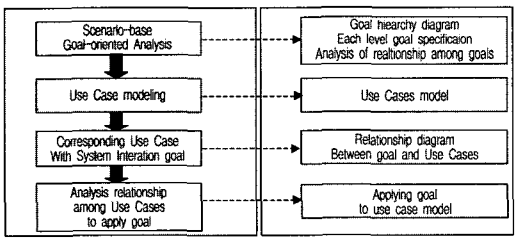
System interaction goal is extracted through the analysis of actor's goal needed to achieve using the system. Use case is elicited from aspects related to the user's usage of system. So, the usage aspect, which is essential to achieve system interaction goal, is appropriate for the use case.

• **Use Case and System internal goal**

After analyzing each use case, it is augmented with internal activity elements of software-to-be. A system internal goal represents system's internal behaviors. It is a set of activities and states to achieve, identified by the system interaction goal. Accordingly, achieving or maintaining system internal goal requires use case to be performed.

3.2 The process for scenario based goal-oriented approach for use case modeling

As previously stated, use case model has some problems. To resolve those problems, we propose the processes of applying goal-oriented analysis to use case model as shown in [Figure 8].



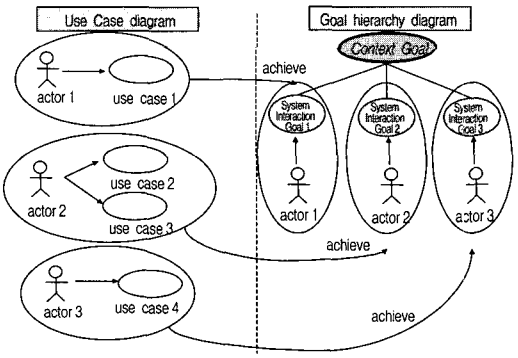
[Figure 8] The process for scenario based goal-oriented approach for use case modeling and its products

At the first step, requirements are analyzed through goal-oriented analysis based on scenarios. We can get goal-hierarchy diagrams, each level's goal-specifications, and results of impacts analysis among goals.

At the next step, the modeling of use cases is performed. At this phase, a use case model is produced.

The next phase is finding what use case corresponds to the system interaction goal. System interaction goals and use cases are elicited on the basis of actors.

Consequently, correspondences of system interaction goals and use case are made on the basis of actors. [Figure 9] shows the approach for making the correspondence between system interaction goals and use case.



[Figure 9] The way of make correspondence of system interaction goals and use case

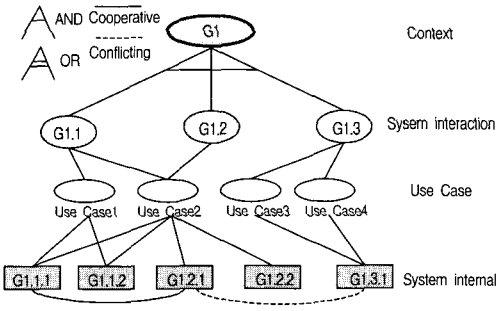
Each use case is connected to the system interaction goal. Use case is also connected to the system internal goal that pertains to the system interaction goal. The diagram produced at this step is shown in section 3.1.

In the next phase, impacts among use cases are analyzed on the basis of goals. Each diagram that represents relationship of goal and use case helps the analyst to write use case model applied goal-oriented analysis. Use case model applied goal-oriented analysis is shown as [Figure 10].

In [Figure 10], one can see that Use Case1 and Use Case 2 are cooperative. On the other hand, Use Case 2 and Use Case 4 are conflicting. Each use case is associated with the goal which can be rationale for requirements, and that is structured. Non-functional requirements that occur in the system internal goal are also expressed.

You can find the reason why requirements are

needed by means of associated goals in the use case model applied goal-oriented analysis. You can also find the relationships among use cases through the analysis of relations of goals. In the model, nonfunctional requirements are represented through associating them with system internal goals.



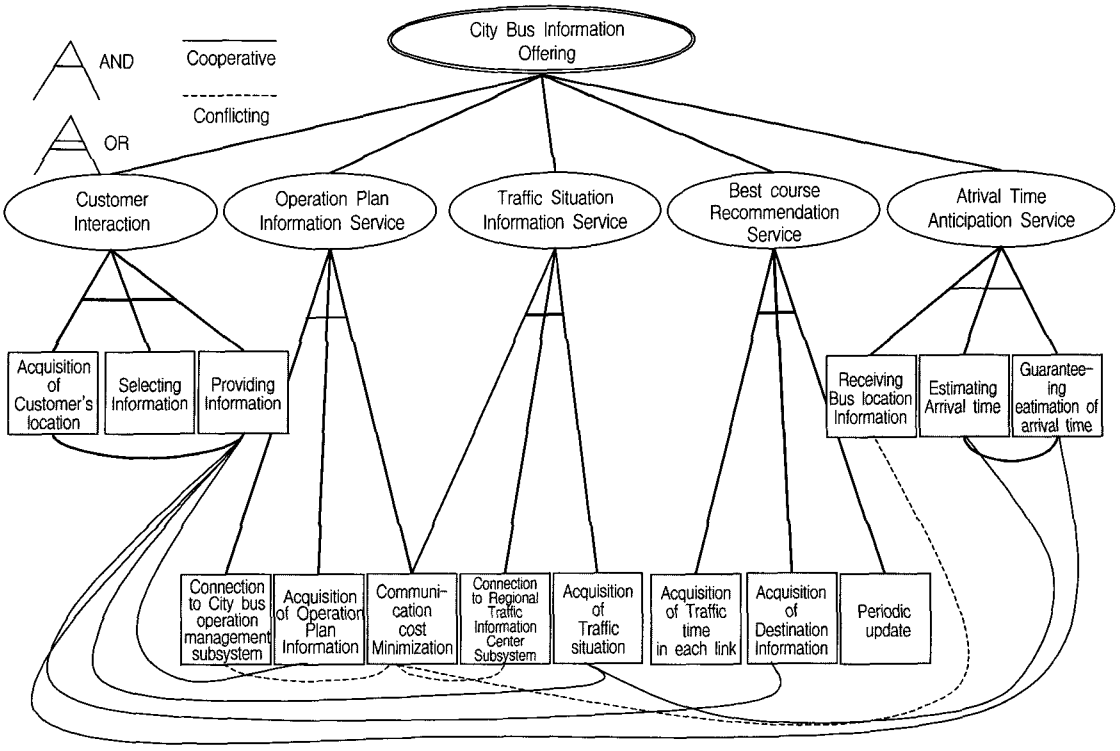
[Figure 10] Use case model applied goal-oriented analysis

4. Case Study : CBIS (City Bus Information Systems) domain

We have applied the proposed goal-oriented analysis method and approach to Use case model to CBIS in the ITS (Intelligent Transport Systems) architecture. CBIS provides bus operation plan information, bus operation situation information, and bus static/dynamic traffic information to customers using a personal device or public device.

4.1 Scenario-based goal-oriented analysis

The analysis has been carried out according to the process stated in section 2.4. Through this process, we can get the goal hierarchy diagram,



[Figure 11] CBIS (City Bus Information Subsystem) domain's goal hierarchy diagram

each level's goal specification, and the result of impacts analysis among goals. Goal hierarchy diagram of CBIS (City Bus Information Subsystem) domain is shown in [Figure 11].

Goal specification of the elicited goals is written out. Through impact analysis of elicited goals, we set up relationships between goals. The result of impact analysis between "Receiving bus location information" goal and "Communication cost minimization" goal can be seen in <Table 4>.

〈Table 4〉 The Side-effect relationship predicates

- ***CP(G[Receiving bus location information], G[Communication cost minimization])*** = $CP_{G[Operation\ plan\ information\ service]}$ ***(G[Receiving bus location information], G[Communication cost minimization])*** $CP_{G[Arrival\ time\ anticipation\ service]}$ ***(G[Receiving bus location information], G[Communication cost minimization])*** = ***FALSE***
- ***CF(G[Receiving bus location information], G[Communication cost minimization])*** = $CF_{G[Operation\ plan\ information\ service]}$ ***(G[Receiving bus location information], G[Communication cost minimization])*** $CF_{G[Arrival\ time\ anticipation\ service]}$ ***(G[Receiving bus location information], G[Communication cost minimization])*** = ***TRUE***

The relationship between "Receiving bus location information" goal and "Communication cost minimization" goal is conflicting. The value of CP (G[Receiving bus location information], G [Communication cost minimization]) is "false", and that of CF (G[Receiving bus location information], G[Communication cost minimization]) is "true". Consequently, you can get the following result as shown <Table 5>.

〈Table 5〉 Relationship between goals

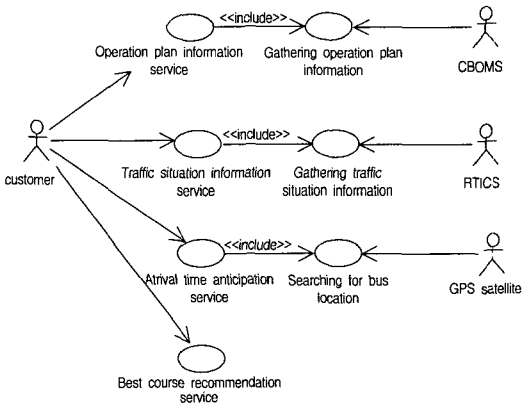
- ***R(G[Receiving bus location information], G[Communication cost minimization])*** = ***<FALSE, TRUE>***

Goals G[Receiving bus location information] and G[Communication cost minimization] are connected in conflicting mode.

"Receiving bus location information" goal and "Communication cost minimization" goal are conflicting. This relationship is reflected in the goal hierarchy diagram shown in [Figure 11].

4.2 Use Case driven analysis

We analyze the problem domain and create a Use case model using Use case in UML. A Use case is a system usage scenario characteristic of specific actor. It represents a usage situation where one or more actors with the aim to accomplish one or more goals use one or more services of the target system. [Figure 12] shows Use case model of CBIS.



[Figure 12] Use case model of CBIS

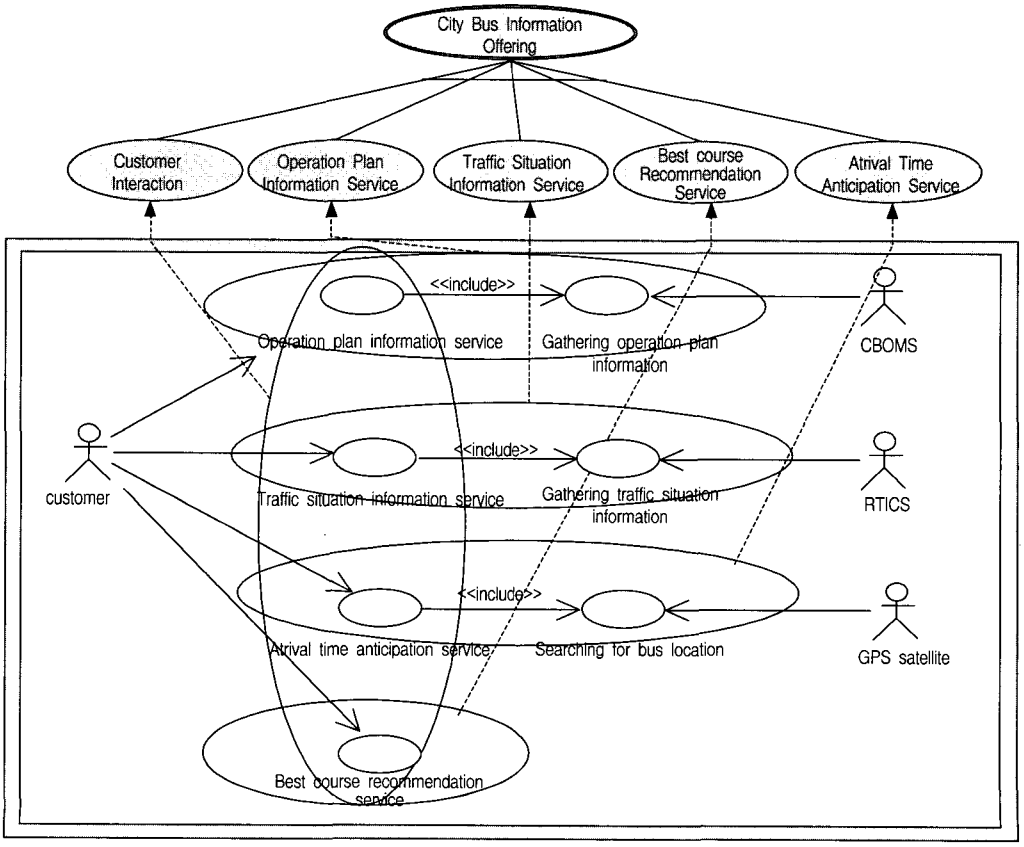
4.3 Correspondence of System interaction goal and Use case

We create correspondences between system

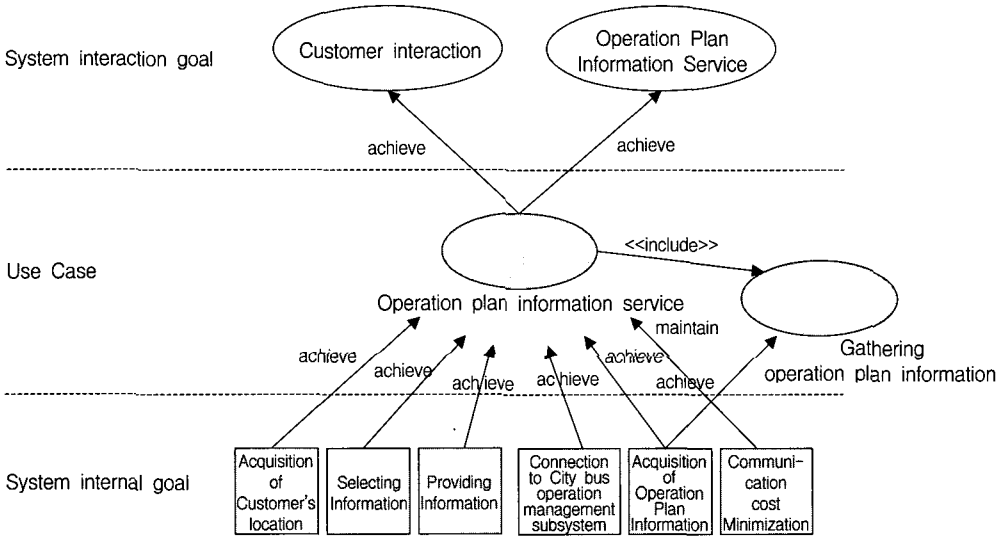
interaction goals and use cases using goal hierarchy diagram and Use case diagram. This correspondence is on the basis of an actor. [Figure 13] shows correspondences of system interaction goals and use cases in CBIS.

After we establish correspondences of system interaction goals and use cases, we generate the relationship diagram between goals and use case. Typically, a use case corresponds to system internal goal through this diagram. [Figure 14] shows the relationship diagram between goal and use case for use cases "Operation plan information service" and "Gathering plan information."

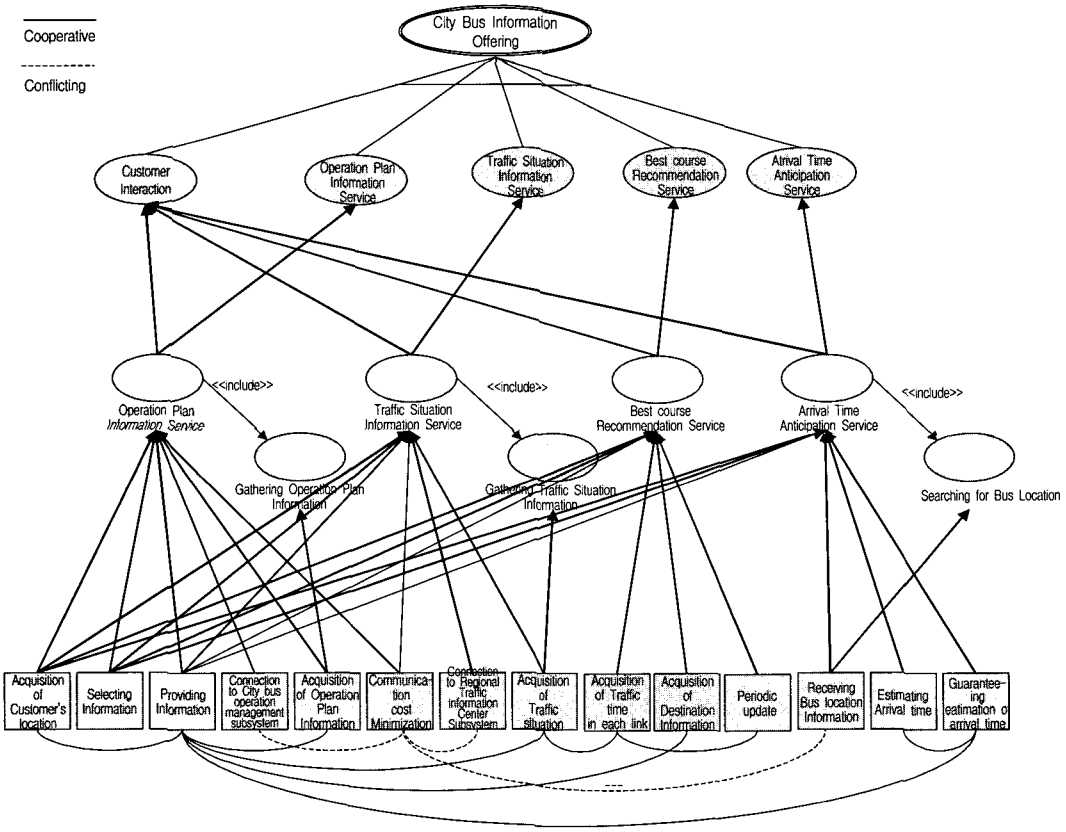
To accomplish the use case "Operation information service" is to achieve system interaction goals "Customer interaction" and "Operation plan information service." To achieve system internal goals "Acquisition of customer's location", "Selecting information", "Providing Information", "Connection of city bus operation management subsystem" and "Acquisition of operation plan information" accomplishes use cases "Customer interaction" and "Operation information service." And to maintain system internal goal "Communication cost minimization" accomplishes use case "Operation information service." We make use case correspond to system interaction goal and create the relationship diagram between goals and use case.



[Figure 13] Correspondences of system interaction goals and use cases in CBIS



[Figure 14] Relationship diagram between goal and use case for use cases "Operation plan information service" and "Gathering plan information"



[Figure 15] The model of the problem domain

4.4 Relationship analysis among Use cases based on goal-oriented analysis

On the basis of the relationship diagram between goal and use case, we write the use case model applied goal-oriented analysis. [Figure 15] is the model of the problem domain.

In [Figure 15], one can see the impacts among use cases in that model. The following table represents side-effect relations among use cases of City Bus Information Subsystem through the use case model applied goal-oriented analysis.

System internal goal	Use Case
Acquisition of Traffic situation	Traffic Situation Information Service
Acquisition of Traffic time in each link	Best course Recommendation Service

System internal goals, "Acquisition of Traffic situation" and "Acquisition of Traffic time in each link", are connected in cooperative mode. R(G[Acquisition of Traffic situation], G[Acquisition of Traffic time in each link]) = (TRUE, FALSE). Consequently, the following use cases "Traffic Situation Information Service" and "Best course Recommendation Service" are connected in cooperative mode.

System internal goal	Use Case
Communication cost Minimization	Operation Plan Information Service, Traffic Situation Information Service
Receiving Bus location information	Arrival Time Anticipation Service

System internal goals, "Communication cost Minimization" and "Receiving Bus location information", are connected in conflicting mode. R(G[Communication-cost-Minimization], G[Receiving-Bus-location-information])-=-(FALSE,TRUE). Accordingly, the use cases, "Operation Plan Information Service" and "Arrival Time Anticipation Service", are connected in conflicting mode. And use cases, "Traffic Situation Information Service" and "Arrival Time Anticipation Service", are also connected in conflicting mode.

5. Related work

5.1 Goal-based reasoning

Prior research efforts in requirement engineering do not capture why a particular set of requirements were included to begin with and whether they were sufficient for achieving the higher-level objective. To resolve these limitations, presently, research on goal-based analysis is being carried out. Yue was probably the first to argue that the integration of explicit goal representations in requirement models provides a criterion for requirements completeness. Broadly speaking, a goal corresponds to an objective the system should achieve through cooperation of agent in software system [9]. Lamsweede and his colleague in goal-directed requirements acquisition presented an approach to requirements acquisition that is driven by such higher-level concepts. These requirements acquisition processes correspond to particular way of traversing the meta-model graph to acquire appropriate instances of the various nodes and links according to such constraints [4]. But it is not easy for an analyst to extract goals from requirements and to depend intensely on KAOS using this method.

5.2 Scenario-based requirement analysis

Several interpretations of scenarios have been presented ranging from examples of behavior drawn from use cases [1], descriptions of system usage to help understand software system [10], and experience based narrative for requirements elicitation and validation [11]. Scenarios are effective means of communicating between users and analysts and supporting requirements analysis. Additionally, scenarios may be used to validate requirements, as 'test data' collected from the observable practice, against which the operation of a new system can be checked. Scenarios may be seen as pathways through a specification of system usage, and represented as simulations of the new system. In order to help analyst extract goals from requirements, we use scenarios as the basis. The problem with scenarios is that they are inherently partial; they raise a coverage problem similar to test case. Scenarios are generally procedural, thus introducing risks of over-specification.

6. Conclusion

Use case driven analysis is widely used in OOAD and CBD. Use case approach looks at the interactions of a single category of users at a time, reducing the complexity of requirements determination. But current use case approaches are somewhat limited in representation of non-functional requirements, in model structuring, and impacts analysis. To alleviate these limitations, we have proposed an approach to apply goal-oriented analysis method to use case model. Goal-oriented analysis is not appropriate for elicitation of the goal and depends on heuristic method. To resolve current goal-oriented analysis method's problems, we have proposed scenario-based goal-oriented analysis method that is easy to elicit the goal and has clear analysis process. Our proposed analysis method helps an analyst to elicit and structure goals and offers a guide to scenario-based goal-oriented analysis method. Through effect relationships, the rationale for designing component in design phase is offered and by recognizing conflicting impacts during early phase, development cost is cut down. We have applied our approach to CBIS and showed how to model conflicting and cooperative relations. For example, we could find 2 cooperative and 3 conflicting relations and also represented 3 non-functional requirements. We plan to continue our research on the tool supporting scenario-based goal-oriented analysis method and its framework.

REFERENCES

[1] Jacobson, I., et al. Object-Oriented Software Engineering, A Use Case Driven Approach, Addison-Wesley, 1992.

[2] Wojtek Kozaczynski, Greedy Booch, "Component-Based-Software Engineering," IEEE Software, 1998.

[3] Colette Rolland, Carine Souyet, Camille Ben Achour Guiding, "Goal Modeling Using Scenarios," *IEEE Transactions on Software Engineering*, Vol.24, No.12(1998), pp. 1055-1071.

[4] Anne Dardenne, Axel van Lamsweerde, stephen fickas, "Goal-directed Requirement acquisition," science of Computer Programming, Vol.20(1993), pp.3-50.

[5] Annie I. Anton, "Goal-Based Requirements

Analysis," Proceedings of ICRE 96, (1996), pp.136-144.

[6] Axel van Lamsweerde, "Requirement Engineering in the Year00 : A Research Perspective," Invited Paper for ICSE '2000 - 22nd International Conference on Software Engineering, Limerick, ACM Press, 2000.

[7] Gerald Kotonya, Ian Sommerville, Requirements engineering : processes and techniques, Wiley, 1998.

[8] Kenha Park, Jintae Kim, Sooyong Park, Goal Based Agent-Oriented Software Modeling, APSEC '00, (December 2000), pp.320-324.

[9] K. Yue, "What Dose It Mean to Say that a Specification is Complete?," Proc. IWSSD -4, Fourth International Workshop on Software Specification and Design, Monterey, 1987.

[10] M. Kyung, "Creating Contexts for Design," J.M. Carroll, ed., Scenario-Based Design : Envisioning Work and Technology in System Development. New York : John Wiley & Sons, 1995.

[11] C. Potts, K. Takahashi, and A.I. Anton, "Inquiry-Based Requirements Analysis," *IEEE Software*, Vol.11, No.2(1994), pp.21-32.