A Goal Modeling Methodology based on High-level Architecture Model for CPS

Minyoung Lee[†, ††], Jeongmin Park[††], Wontae Kim[††] and Ingeol Chun[†, ††]

[†]Department of Computer Science and Engineering, University of Science & Technology, Daejeon, Republic of Korea
[††]Department of Embedded SW Research, ETRI, Daejeon, Republic of Korea

Summary
Cyber-Physical Systems (CPS) is computing system integrating with physical elements including sensors, actuators and computing elements controlling these physical environments. In CPS operating environments, CPS is required to provide high-reliable services responding to the complexity and uncertainty of the surrounding physical environments. Therefore, autonomic computing technology which manages undesirable situations at the operation time without human intervention is necessary for CPS. For autonomic control, a goal model is a prerequisite. A goal model capturing desirable system runtime states can be facilitated as core knowledge in evaluating system runtime states to adapt to undesirable situations. In this paper, we demonstrate goal model definitions and terminologies required to describe a goal model. And, we review traditional goal model description technologies. Finally, we propose a goal modeling methodology for self-adaptive CPS based on high-level architecture model.

Key words:
CPS, Autonomic Computing, Goal Model, High-level Architecture Model

1. Introduction

As the social infrastructures tend to be controlled by computing devices[1], systems integrating these numerous computing devices without human intervention are increasing. The advent of Cyber-Physical Systems (CPS) has evolved the paradigm of such embedded software development method[2].

CPS is large-scale computing system integrating with physical elements including sensors, actuators and computing elements controlling these physical environments[1]. In CPS operational environment, all components interact with uncertain and unpredictable physical environments. Therefore, CPS is required to provide high-reliable services responding to the complexity and uncertainty of the surrounding physical environments. In this context, autonomic computing technology which manages undesirable situations at the operation time without human intervention is necessary for CPS.

Autonomic computing is a computing technology that suggested in the IBM to support the self-management of enterprise applications[3]. It includes control loop called MAPE-K(Monitor, Analysis, Plan, Execute, Knowledge)[4] which monitors some managed resources (SW or HW components) and autonomously tries to keep their parameters within a desired range in order to control the behavior of the system.

To ensure autonomic control, a goal model is a prerequisite. A goal model has been widely used at requirements analysis phase of traditional software development methodology[5]. A goal captures a system state that a system is supposed to do[6]. There are a number of goal-oriented requirements modeling approaches. But, there aren't enough leading research and actual practice applying a goal modeling approach to desirable system runtime states modeling.

Our approach takes advantages of a goal model to capture desirable system runtime states according to a specific top goal. A goal model can be facilitated as core knowledge in evaluating system runtime states to adapt to undesirable situations. Each goal refers to a specific system runtime state required to ensure high-reliable services. Therefore, autonomic system performs self-management abilities such as self-healing, self-configuring, self-protection, self-optimizing by MAPE control loop in evaluating various system runtime states and adjusting system at the operation time.

Especially, we propose a novel goal modeling approach capturing a desirable system runtime states set based on high-level architecture model.

The reminder of the paper is organized as follows. Section 2 demonstrates goal model definitions and terminologies required to describe a goal model. And, we review traditional goal model description technologies such as KAOS, goal modeling method of an adaptive system based on KAOS, goal tree with fault tree, i* framework, Tropos. Section 3 accounts for the details of our approach. Section 4 demonstrates our approach in applying to self-driving car example. Finally, we present conclusions.

Manuscript received October 5, 2013
Manuscript revised October 20, 2013

2. Related Works

2.1 Goal-Oriented Requirements Engineering

A goal model has been widely used at requirements engineering[6]. In requirements analysis phase, user interests and important concerns are captured. In other words, it is enough to say that a goal captures a system state that a system is supposed to do[6]. In world modeling, it has been found useful to distinguish among three basic ontological concepts: entities, activities, and assertions[7]. Likewise, they can be applied to a goal model. A goal model requires three concepts: Goal, Operation, Condition. A goal as a kind of assertion is a certain state to be achieved or hold. More specifically, a goal refers to as an objective to be achieved[8] or an intended property to be ensured[9]. A goal can be identified by asking how a goal is achieved or why a goal is needed about the goal already described. An operation as a kind of activities is a task required to achieve a goal. A condition as a kind of assertions is a constraint to evaluate a goal satisfaction. Identified concepts in a goal model are structured by links such as AND, OR link[6].

2.2 Goal Model Description Technologies

There are a number of goal-oriented requirements modeling approaches. In approaches, the concept of a goal as a part of requirements engineering technique is used.

The Knowledge Acquisition in autOmated Specification (KAOS)[10] uses the combination of four models: goal model, operation model, object model, responsibility model. The goal of the KAOS is to describe problems and alternatives. But, the KAOS is not intended for code generation due to natural language description. A goal modeling methodology of an adaptive system based on the KAOS[11] is to develop requirements for a DAS considering different environmental conditions. It makes a system reliable by using three uncertainty mitigation tactics: adding a sub-goal or a high-level goal, using a RELAX language. But, this approach is difficult to realize code generation due to natural language description likewise. And, it requires additional goal refinement processes for the mitigated goal. Moreover, it may result in state explosion. A goal modeling methodology with fault tree[12] is developed for diagnosing a problem considering different system environmental information. The transformation from a goal tree to a fault tree supplies analysis for system complexity and a problem as well a root cause of a problem. The i* framework[7] is developed for modeling requirements by representing them as dependencies between actors. The SD model describes strategic dependencies between actors for achieving goals, while the SR model describes intentions for actors and dependencies between actors. This framework lacks concepts of obstacle, event, object and can be complex and huge by refining the model. The tropos[8] is an agent-oriented software development methodology dealing with the whole of software development phases using basic concepts of the i* framework. Therefore, basic concepts of the tropos are very similar to the i* framework except for a plan referring to as a course of tasks. This methodology doesn’t deal with the concept of risk interrupting with achieving the goal.

2.3 Problem Statement

As far as we know, there are no approaches applying a goal modeling approach to desirable system runtime states modeling. In addition, there aren’t desirable system runtime states description technologies considering a high-level architecture model. Therefore, it is difficult to identify the associated entity, relationship, property through a goal model. We propose a novel goal modeling approach capturing a desirable system runtime states set based on a high-level architecture model.

3. A Goal Modeling Methodology based on High-level Architecture Model

CPS ensures high-reliable services responding to the complexity and uncertainty of the surrounding physical environments in exploiting autonomic computing technology. Autonomic computing technology is compared to human autonomic nervous system. Therefore, System itself can monitor system runtime states and adjust abnormal situations. To realize autonomic control of CPS, it is required to define a top goal that the system should achieve. And, desirable system runtime states are necessary for realizing the top goal. Therefore, autonomic computing can be realized in evaluating these system runtime states and adjusting within a desired range.

In this paper, we take advantage of a goal model to represent a top goal. A goal model captures desirable system runtime states according to a specific top goal. Each goal refers to as specific system runtime states required to ensure high-reliable services. By using a goal model, we provide explicit causal relationships on why, how goals are required. And, if a goal model is assigned as core knowledge for autonomic control, it is feasible to reduce much efforts and time on visualizing goals and alternatives.

In addition, we propose a novel goal modeling approach associating a desirable system runtime state with entity, relationship, property in a high-level architecture model. This enables locating the abnormal entity, relationship, property when the associated goal is not achieved.

Therefore, we propose a goal modeling process capturing a desirable system runtime states set based on a high-level architecture model in this section. Prior to starting to the goal modeling process, one of the most important is that a top goal should be defined firstly. A top goal can be a self-healing, self-optimizing, self-configuring, self-protection and so on according to a purpose. The goal modeling process is organized as follows Figure 1.

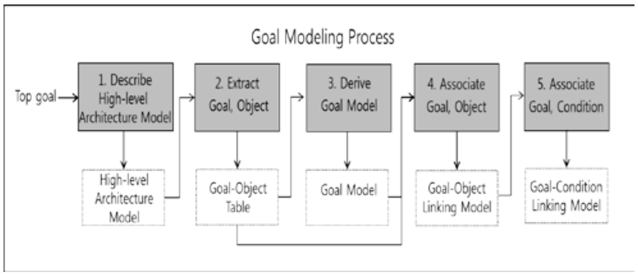

Figure 1 Goal Modeling Process based on a High-level Architecture Model

Step1. Describe high-level architecture model

It is required to describe a high-level architecture model according to the top goal. A high-level architecture model includes system major physical elements and relationships among physical elements according to the top goal. Therefore, it can identify system operating information. In other words, elements in a high-level architecture model are used monitoring system operating information making it possible to result in abnormal runtime states. The example of a high-level architecture model is as follows Figure 2.

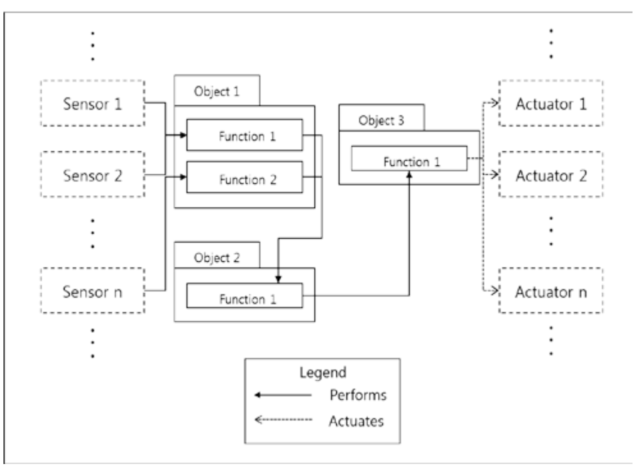

Figure 2 The example of a high-level architecture model

Step2. Extract goal and object

In this step, a goal with associated object extraction process is proceed based on the top goal and the high-level architecture model. A goal refers to as a certain system runtime state relating to the associated object in the high-level architecture model. The extracted goals and the associated objects consist of goal-object table. The example of a goal-object table is as follows Table 1.

Table 1: The example of a goal-object table

Goal	***Object***
G_1	Ob_1, $Ob_{1.1}$, $Ob_{1.2}$, $Ob_{1.3}$
G_2	Ob_1, Ob_2, $Ob_{2.1}$
G_3	Ob_2
G_4	Ob_1, $Ob_{1.1}$, $Ob_{1.2}$, $Ob_{2.1}$, $Ob_{2.2}$

Step3. Derive goal model

Then, a goal model is described by reflecting the causal relationship between goals based on the goal-object table. Therefore, we provide explicit understanding of on why, how goals are required. In addition, a domain property referring to as a domain assumption between G1 and G2 required to achieve G1. The example of a goal model is as follows Figure 3.

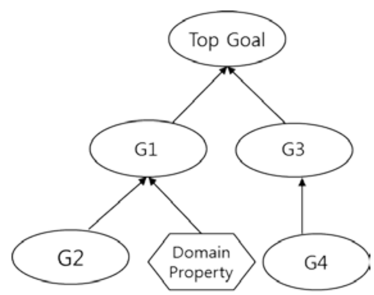

Figure 3 The example of a goal model

Step4. Associate goal and object

In this step, the derived goal model should be associated with objects by using the goal-object table and goal model. Each goal satisfaction is responsible for each associated object in the high-level architecture model according to goal-object table in the step 2. Therefore, associated objects are defined in the goal-object linking model. The example of a goal-object linking model is as follows Figure 4.

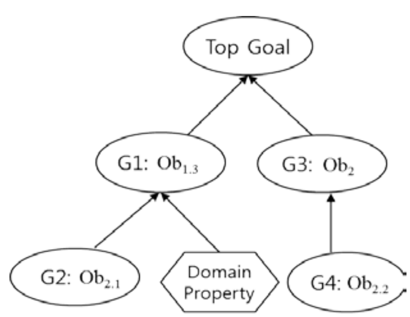

Figure 4 The example of a goal-object linking model

Step5. Associate goal and condition

To evaluate goal satisfaction, a condition should be defined. A condition refers to as whether a goal is satisfied. Therefore, conditions are extracted based on goal-object linking model. The extracted conditions are connected with the associated goals. The example of a goal-condition linking model is as follows Figure 5.

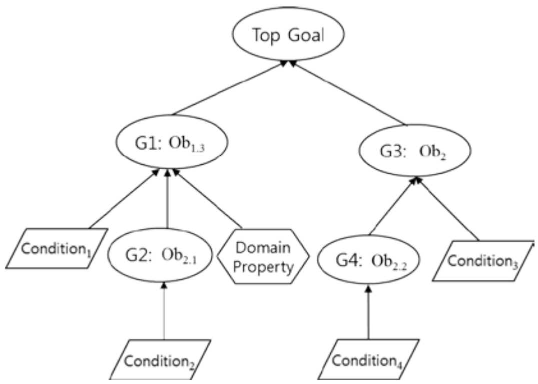

Figure 5 The example of a goal-condition linking model

4. Case Study

In this case study, we describe a goal model as an unmanned self-driving car example. An unmanned car is capable of self-driving without human intervention. It is important to ensure safety driving from a point of departure to destination. Therefore, we define 'Safety Driving' as a top goal of an unmanned self-driving car. The case study is performed by the following 5 steps.

Step1. Describe high-level architecture model

It has four sensors such as laser range finder, position sensor, gyroscope sensor, radar sensor and GPS to collect information about operating car and environmental conditions. Laser range finder sensor measures distance of surrounding objects in the range of the car. Radar sensor measures distance, altitude, direction, velocity of surrounding objects. Position sensor and gyroscope sensor measure the state of balance, the angle of wheel in the car. GPS provides location information. Vision camera identifies environmental traffic information. Therefore, computer in the car decides driving plan by incorporating collected information from four sensors. As well as, computer transmits driving plan signal to car physical devices to adjust car's driving path. The described high-level architecture model of an unmanned self-driving car is as follows Figure 6.

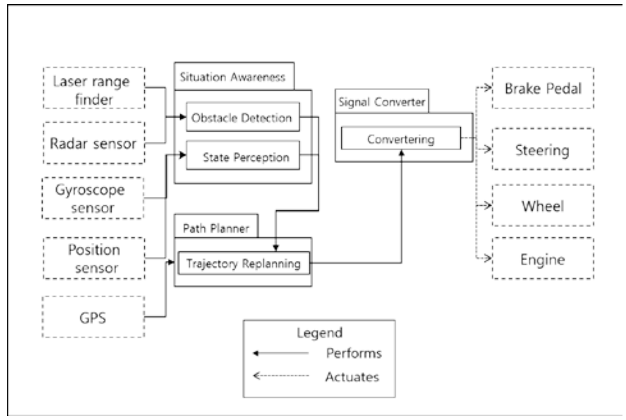

Figure 6 The high-level architecture model of an unmanned self-driving car

Step2. Extract goal and object

In this step, a goal with associated object extraction process is proceed based on the top goal and the high-level architecture model. The goal-object table of an unmanned self-driving car is as follows Table 2.

Table 2: The goal-object table of an unmanned self-driving car

Goal	***Object***
Safe distance between cars	Radar Sensor, Situation Awareness, Path Planner, Signal Converter, Brake Pedal, Wheel
Lane departure avoidance	Laser range finder, Gyroscope Sensor, Situation Awareness, Path Planner, Signal Converter, Steering, Wheel
Radar sensor normal operation	Radar Sensor
Safe velocity	Radar Sensor, Situation Awareness, Brake Pedal, Wheel
Steering command normal operation	Laser range finder, Gyroscope Sensor, Situation Awareness, Steering, Wheel

Step3. Derive goal model

Then, a goal model is described by reflecting the causal relationship between goals based on the goal-object table. The domain assumption such as 'Steering observation 10 times per 1 second' is added between the goal 'Lane departure avoidance' and the goal 'steering command normal operation' required to achieve goal 'Lane departure avoidance'. The steering control should be checked 10 times per 1 second. Therefore, the steering control and the sensing information from gyroscope sensor can be checked whether they are matched. The goal model of an unmanned self-driving car is as follows Figure 7.

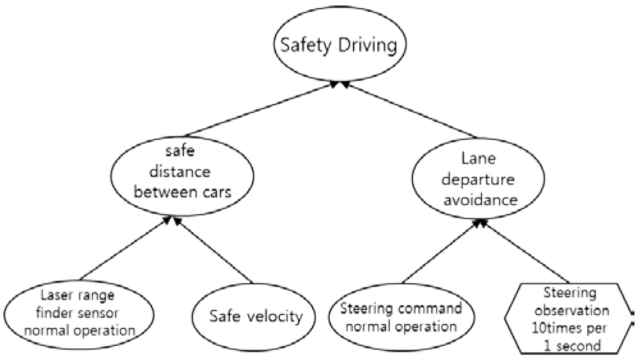

Figure 7 The goal model of an unmanned self-driving car

Step4. Associate goal and object

The derived goal model is associated with objects by using the goal-object table and goal model. Therefore, the goal-object linking model is acquired. The goal-object linking model of an unmanned self-driving car is as follows Figure 8.

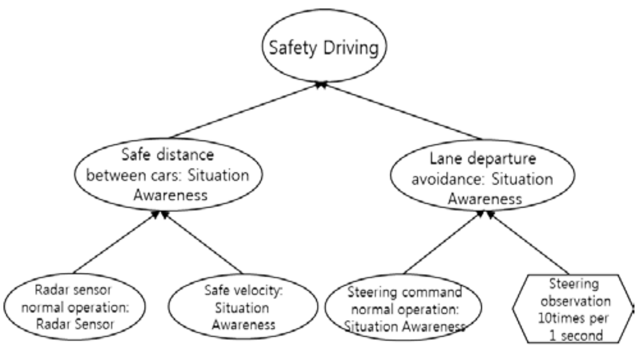

Figure 8 The goal-object linking model of an unmanned self-driving car

Step5. Associate goal and condition

Conditions are extracted based on goal-object linking model. The extracted conditions are connected with the associated goals. The goal-condition linking model of an unmanned self-driving car is as follows Figure 9.

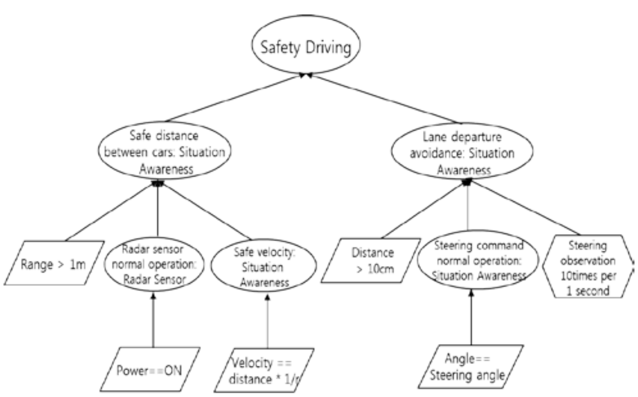

Figure 9 The goal-condition linking model of an unmanned self-driving car

5. Conclusions

In this paper, we have demonstrated the goal model definitions and terminologies required to describe a goal model. And, we have reviewed goal model description technologies. Finally, we have proposed a goal modeling methodology based on a high-level architecture model for CPS. If a goal model is assigned as core knowledge for autonomic control, it is feasible to reduce much efforts and time on visualizing goals and alternatives. In addition, our approach enables locating the abnormal entity, relationship, property when the associated goal is not achieved by associating a goal model with a high-level architecture model. In future work, we are planning to add property referring to as CPS characteristics. In addition, goal satisfaction evaluation method will be specified considering the added property.

Acknowledgment

This work was supported by the IT R&D Program of MSIP/KEIT [10035708, "The Development of CPS(Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software"]

References

[1] Edward A. Lee, “Cyber Physical Systems: Design Challenges”, In 11th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing(ISORC), pp.363-369, IEEE, May, 2008
[2] E. A. Lee, "Cyber-Physical System-Are Computing Foundations Adequate?", Proc. NSF Workshop on cyber-Physical Systems, 2006
[3] IBM, “An Architectural Blueprint for Autonomic Computing 3rd Edition”, IBM Autonomic Computing White Paper, pp.3-31, 2005
[4] S Dobson, “A Survey of autonomic computing-degrees, models and applications”, 2006
[5] Chonghyun Lee, Hyunsang Youn, Ingeol Chun and Eunseok Lee, “A Runtime Evaluation Methodology and Framework for Autonomic Systems”, Proceedings of International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2011
[6] Axel van Lamsweerde, “Goal-oriented requirements engineering: a guided tour”, Proceedings Fifth IEEE International Symposium on Requirements Engineering, 2001
[7] Eric S.K. Yu, “Modeling Strategic Relationships for Process Reengineering”, Ph.D Thesis, Graduate Department of Computer Science, University of Toronto, 1995
[8] Bresciani P, Giorgini P, Giunchiglia F, Mylopoulos J, Perini A, “The Tropos: An Agent-oriented Software Development Methodology”, Technical Report, ITC-irst, December, 2002
[9] Mirko Morandini, Loris Penserini, Anna Perini, “Towards Goal-Oriented Development of Self-Adaptive Systems”, In Proceedings of the 2008 international workshop on software

engineering for adaptive and self-managing systems, pp. 9-16, 2008
[10] Lamsweerde, A. "Kaos tutorial" Cediti, September 5, 2003
[11] Betty H.C. Cheng et al., “A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty”, LNCS, 2009
[12] Qin Zhu, “Goal trees and Fault trees for root cause analysis”, In proceedings of the 2008 international conference on software maintenance, pp. 436-439, 2008