

Goal Modeling Method for Autonomic Control Of Cyber-Physical Systems (CPS)

Jeongmin Park*, Sungjoo Kang*, Ingeol Chun*, Wontae Kim*

*Embedded SW Department, Electronics and Telecommunications Research Institute (ETRI),

218 Gajeongno, Yuseong-gu, Daejeon, Korea

jmpark23@etri.re.kr, sjkang@etri.re.kr, igchun@etri.re.kr, wtkim@etri.re.kr

Abstract — This paper presents an approach to implementing ‘Goal Model’ for autonomic control of Cyber-Physical Systems (CPS). The approach presents to define and apply system knowledge required for autonomic control from goal model.

Keywords— CPS, Autonomic Control, Goal Model, Self-Adaptation, Embedded SW

I. INTRODUCTION

Cyber-physical systems (CPS) are an integrated system that involves network, software and physical system including sensors and actuators [1]. In general, we are largely able to divide CPS into two types. The first part is the physical world that manages physical environment or object required in real world. The second type is the cyber space that retains environment operated by software. Therefore, CPS is an integrated system including ‘Hardware’, and ‘Software’.

However, as management complexity and maintenance cost of CPS keep spiraling upward, it is a critical and difficult job for humans to maintain and manage CPS in a complicated computing environment. Actually, it is very difficult to manage and maintain CPS through human manual operation. Especially, it is hard to recognize the diverse problems incurred in CPS and solve the recognized problems. Humans directly managing CPS should be experts. This imposes limitations due to the availability of human resources and constrains effective cost management [2-3].

As way to deal with these problems, ‘Self-Adaptation’ research on high-reliability for CPS is recently hot issue. ‘Self-Adaptation’ approach has the issues comprising 1) goal modelling, 2) monitoring, 3) analysis, 4) planning, and 5) execution. An important thing among these issues is ‘Goal Modeling’, because ‘Goal Model’ is an essential knowledge facilitating ‘Self-Adaption’ and abstracting the complicated management problems.

‘Goal Model’ is a model that describes ‘Goal’ to be achieved in Cyber-Physical System (CPS). That is, goal model means core knowledge that ensures high-reliability to be retained in CPS. Thus, this paper presents an approach to generating ‘Goal Model’ for autonomic control that spontaneously offsets the problems incurred from CPS. This approach consists of various steps such as 1) goal extraction, 2) goal interpretation, 3) association between goal tree and

designed model, and 4) linking leaf-goal and constraints. Through these activities, it is possible to help ensure that ‘Goal Model’ is able to become a key knowledge towards autonomic control which adjusts CPS’s abnormal operation.

II. PRELIMINARY RESEARCH AND PROBLEM STATEMENT

A. From Goal Model to Autonomic Control System

In order to address what ‘Goal Model’ is and what ‘Goal Model’ can do for ‘Autonomic Control System’, we will be able to represent that ‘Goal Model’ promises to reduce most of the complexity and software maintenance costs from CPS [4].

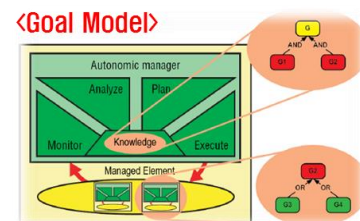


Figure 1. Architecture and Goal Model for Autonomic Control system

Fig.1. shows architecture of ‘Autonomic Control System’ (This architecture was depicted by [4]). ‘Autonomic Control System’ is architectural component called Autonomic Manager that controls and manages elements such as resources, components, etc. In order to accomplish it, ‘Autonomic Control System’ have to be instrumented with tools for monitoring managed elements, for analyzing the collected data to determine whether managed elements are performing as expected, for planning a strategy regarding detected problem, and for executing planning strategy, these processes solicit the goal of the ‘Autonomic Control System’.

B. Problem Statement

There are few related works and applicable cases for autonomic control of CPS, because the research is still initial stage. But there are three basic ways to make a ‘Autonomic System’. The first way is to build planning capabilities and self-configuration skills that reconfigures task through external system components [5]. The second way is artificial intelligence approaches to autonomic systems [6]. The third

way is of part for ensuring high-reliability of CPS. These approaches have been proposed to develop autonomic control system including ‘Reliable Home Service Robot [7]’, ‘Vehicular Cyber-Physical System [8]’, ‘Self-Adaptive Robot [9]’ in CPS research community. They have been developed to make the systems more reliable. On one hand, most of the existing research has common problems that developers have to analyze the complex target CPS in person. Namely, it requires much effort and time inspecting it. If it is possible to assign core knowledge for autonomic control using ‘Goal Model’, it is feasible to reduce much efforts and time. In this paper, we consider the collaboration between designer and developer, and propose an approach to generating ‘Goal model’ for autonomic control. Through the proposed factors, it is possible to recover abnormal problems and to operate them safety.

III. GOAL MODELING REQUIREMENTS FOR AUTONOMIC CONTROL

In this section, we briefly describe requirements of ‘Goal Modeling’ for ‘Autonomic Control’ that can adapt to changes in external and internal situation. These requirements are demonstrated in Table I.

TABLE 1. GOAL MODEL REQUIREMENTS

Phase	Description
Step1	Goal Extraction
Step2	Goal Interpretation and Goal Tree Generation
Step3	Linking Constraints
Step4	Setting Autonomic Control Point

In order to satisfy ‘Goal’ to be achieved in CPS environment, as described in Table I, There are four easy ways for modeling ‘Goal’ to make ‘Autonomic Control System’. Firstly, step1 is to extract ‘goal’ to use the natural language. Secondly, step2 is to interpret the extracted goal by analyzing ‘Goal_ID’, and generates ‘goal tree’ to comprise an appropriate data structure. Step3 is to associate the goal tree with designed model for mapping the goal tree and sub-modules generated in design time, and then to link constraints to ensure normal systems state on goal tree. Finally, step4 is to set autonomic control points if constraints are violated.

‘Autonomic Control System’ has the capabilities such as self-configuration, self-healing and self-optimization by having the ability to manage, and reduce the complexity and maintenance costs of CPS. The ability of these ‘Autonomic Control’ analyzes and changes the behavior of the various systems in run-time. Therefore, ‘Autonomic Control System’ necessarily needs to have ‘Goal Model’ to achieve the management purpose of ‘target system’. ‘Goal Model’ is the core knowledge of ‘Autonomic Control System’ [1].

IV. CPS ENVIRONMENT

This section briefly introduces an ‘Intelligent Service Robot (ISR)’ accomplished as previous research [10]. Through this ISR (see Fig.2), we are going to apply ‘Goal Model’.

A. Intelligent Service Robot for CPS

As presented in Fig.2, ISR is service robot that has various sensors and carries out the tasks if it is commanded to do. User application operates interacting with ISR. In here, we will be able to explain ‘what is CPS?’. User Application means ‘Cyber Space’ that retains environment operated by software. ISR is in the real world. We call these CPS.



Figure 2. User Application and ‘Intelligent Service Robot(ISR)’

B. Scenario

The action scenario of the user application is presented in Fig. 3. When the user draws the desired zone or room by pushing the button in user application, user application creates the drawing path, and sends the route to ISR. The web camera sends the images around ISR to the web server. Then, through the internet, user application shows the images to the user.

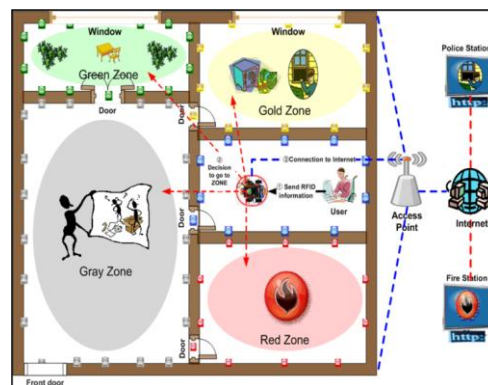


Figure 3. Scenario of User Application

As presented in Fig.4, CPS is consisting of user, service robot, user interface, database server, and web camera. Web server installed, which enables it to be access from the outside, and web server which saves log data using DB server (see Fig.4). User application sends user’s commands to the robot and shows the information. It communicates with the robot via bluetooth and is linked with the internet through a web server. User application has two modes: drawing mode in which the user can directly draw the navigation zone and click mode in which the user can select the room that the robot is to navigate

to by pushing the buttons designated for each room.



Figure 4. Architecture of CPS

Once navigation begins following the user's command, the route from the current location of the robot to the final destination is determined and sent to the robot. After the robot starts making its way to the destination, the location of the robot can be tracked using the ID number of the RFID Tag that is relayed back.

V. APPLYING GOAL MODEL TO CPS ENVIRONMENT

In this section, we briefly apply 'Goal Model' to an 'Intelligent Service Robot (ISR)'. To illustrate what goal model are, and what they can do for the design of autonomic control for ISR, we suppose the task that ISR have to achieve a goal that ascertains 'Driving straight'.

A. Step1: Goal Extraction

Goal Extraction step is to extract goal achieved in ISR using the natural language. Fig.5 is to explain Goal Extraction

System's Goals	
G1	Driving Straight
G2	Angle variation of compass sensor is 0(zero)
G3	Arriving within 10 seconds
G4	The number of rotation of motor has to be consistent
G5	...
G6	...
GN	...

Figure 5. Goal Extraction

B. Step2: Goal Interpretation and Goal Tree Generation

Goal Interpretation and Goal Tree Generation step is to interpret the extracted goal, and generates goal tree according to dependencies of goal IDs such as G1, G2, and etc. Fig.6 is goal tree generated by Step1 that is Goal Extraction.

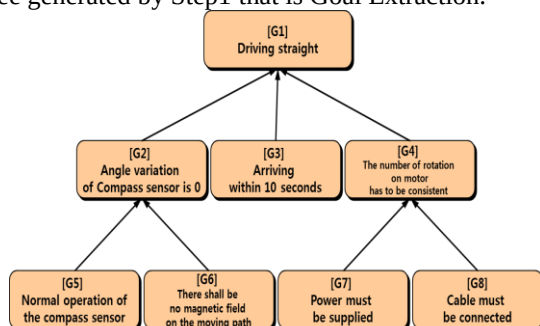


Figure 6. Goal Tree generated based on Step1

C. Step3: Linking Constraints

Goal Linking Constraints step is consisted of two parts. Firstly, it is to associate the goal tree with designed model for mapping sub-modules generated in design time. It is depicted in Fig.7 as below.

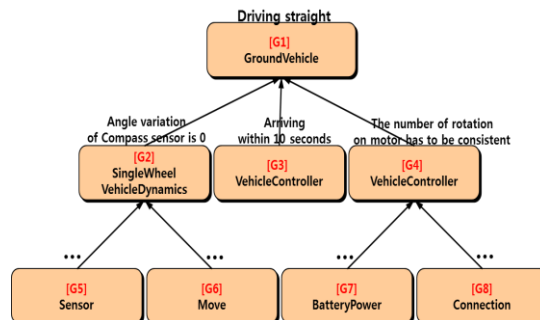


Figure 7. Mapping Goal Tree and Sub-Modules

Secondly, it is to link constraints to ensure normal systems state on goal tree. Leaf-node linked with constraints is not sub-module but system-state or sensor name. Fig.8 is to link constraints and goal-tree.

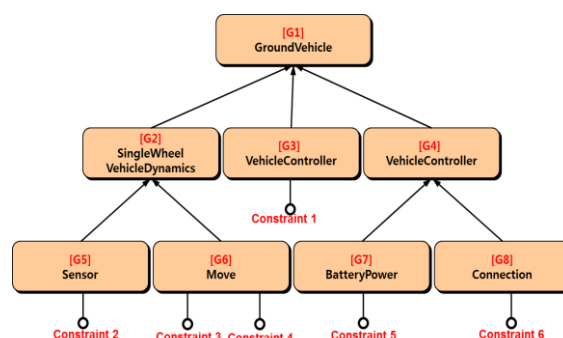


Figure 8. Linking constraints and goal tree

For example, [G7]'s constraint, constraint5 means that 'BatteryPower' must be supplied. Constraint6 explain that cable must be connected.

D. Step4: Setting Autonomic Control Point

Setting Autonomic Control Point step is to identify the violated constraints as 'Autonomic Control Point (ACP)'. It is imperative to detect and identify ACP. This means that detecting the ACP is to alleviate management complex of CPS and also make realize to necessitate 'Autonomic Control System'. Fig.9 show example detecting violated constraints.

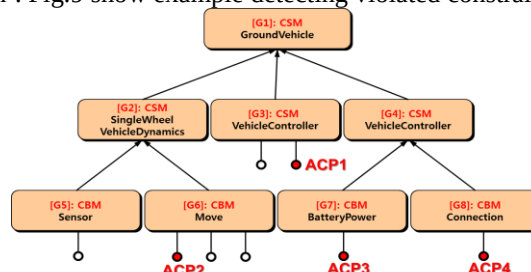


Figure 9. Setting Autonomic Control Pointer(ACP)

VI. EVALUATING GOAL ACHIEVEMENT

In this section, we evaluate ‘Goal Achievement’ applied to an ‘Intelligent Service Robot (ISR)’. In order to evaluate ‘Goal Model’ applied to ISR, as Fig.10, we included the following information. Firstly, Goal ID is identifier (i.e. G1, G1.1, G1.2, ..., G1.3.2). Goal Name is name of goals to be achieved. Weight (w) is value to be accomplished for each of goals preheated by designer. Mandatory (M) is inevitably to satisfy the mission.

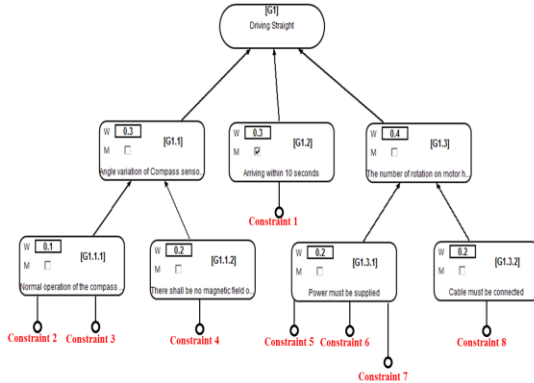


Figure10. Goal Model applied to ‘ISR’

Constraint's Name of Object1 is Constraint1.....inspecting
Result: 1
Success.....<NORMAL STATUS>
Constraint's Name of Object1 is Constraint2.....inspecting
Result: 1
Success.....<NORMAL STATUS>
Constraint's Name of Object1 is Constraint3.....inspecting
Result: 1
Success.....<NORMAL STATUS>
Constraint's Name of Object1 is Constraint4.....inspecting
Result: 1
Success.....<NORMAL STATUS>
Constraint's Name of Object1 is Constraint5.....inspecting
Result: 1
Success.....<NORMAL STATUS>
Constraint's Name of Object1 is Constraint6.....inspecting..
Result: 0
Failure.....<ABNORMAL STATUS>
Constraint's Name of Object1 is Constraint7.....inspecting
Result: 1
Success.....<NORMAL STATUS>
Constraint's Name of Object1 is Constraint8.....inspecting
Result: 1
Success.....<NORMAL STATUS>

Figure11. Setting autonomic control pointer

We implemented an ‘ETRI-detector’ as prototype for evaluating the goals. It inspects mapped constraints on goal tree whether there is an abnormal state or not. Fig.11 presents these activities. Violated constraint is the constraint6. The weight of constraint6 is about 0.07 (that is 0.2 / 3).

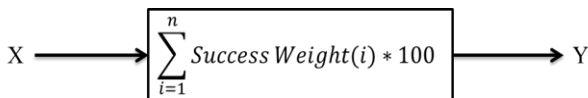


Figure12. Function calculating goal achievement ration

Using function depicted in Fig.12, we compute our goal model’s achievement evaluating success ratio of constraints. $Y = (0.3 + 0.05 + 0.05 + 0.2 + 0.07 + 0.07 + 0.2) * 100$. Thus, Y is about 94% as goal achievement. In Fig.13, state monitor presents red color as evidence of problem arising in CPS environment.



Figure13. State monitor in User Application

VII. CONCLUSION

This paper proposed method modeling goal for autonomic control of CPS. In future work, we will make extended research through proposed goal model.

ACKNOWLEDGMENT

This work was supported by the IT R&D Program of MKE/KEIT [10035708, “The Development of CPS(Cyber-Physical Systems) Core Technologies for High Confi-dential Autonomic Control Software”].

REFERENCES

- [1] J.M. Park, S.J. Kang, I.G. Chun, and W.T. Kim "An Approach to Generating Goal Model for Cyber-Physical Systems," In Proc. International Symposium on Embedded Technology, Daegu, Korea, pp. 23-24, May, 2013.
- [2] J.O. Kephart, and D.M.Chess "The vision of autonomic computing," IEEE Computer Society, vol.36, pp.41-50, January, 2004.
- [3] L. Rechard. "Creating robust software through self-adaptation". IEEE Intelligent Systems, vol.15, pp.26-29, May, 2000
- [4] A. Lapouchnian, Y.Yu, S. Liaskos, and J.Mylopoulos " Requirements-Driven Design of Autonomic Application Software," In Proc. DEAS 2005, LA, USA, pp.20-26, May, 2005.
- [5] J.H. Lee, J.M. Park, G.J. Yoo, and E.S. Lee "Goal-Based Automated Code Generation in Self-Adaptive System," Journal of Computer Science and Technology, vol. 25, pp.1118-1129, Nov. 2010.
- [6] J.M. Park, H.S. Youn, and E.S. Lee " An automatic code generation for self-healing," Journal of Information Science and Engineering, vol. 25, pp.1753-1781, Nov. 2009.
- [7] I.G. Chun, H.Y. Lee, W.T. Kim and E.S. Lee, "SMART-CPS: Self-Managed Reliable System Development Method for Cyber-Physical Systems", Communications in Computer and Information Science, Vol. 256, pp.138-148, Nov. 2011.
- [8] X. Li, C. Qiao, X. Yu, A. wagh, R.Sudhaakar, and S. Addepalli, "Toward Effective Service Scheduling for Human Drivers in Vehicular Cyber-Physical Systems", IEEE Transactions On Parallel and Distributed Systems, Vol. 23, No.9, pp.1775-1789, May 2012.
- [9] I.G. Chun, J.M. Kim, H.Y. Lee, and W.T. Kim, "Faults and Adaptation Policy Modeling Method for Self-Adaptive Robots", Communications in Computer and Information Science, Vol. 150, pp.156-164, Sep. 2011.
- [10] J.M. Park, I.G. Chun, H.Y. Lee, W.T. Kim, and Eunseok Lee. "Intelligent Service Robot and Application operating In Cyber-Physical Environment," Lecture Notes in Electrical Engineering, vol.181, pp.301-310, Sep. 2011.



Jeongmin Park received his Ph.D. and M.S. degrees in Department of Computer Engineering from Sungkyunkwan University, Korea, in 2009 and 2005, respectively, and his B.S. degree in Computer Engineering from Korea Polytechnic University, in 2003. He is currently a senior member of engineering staff in ETRI, Korea. His research interests include Cyber-Physical System (CPS), Autonomic Computing and Software Engineering.



SungJoo Kang received his M.S. and B.S. degrees in Electronic Engineering from Hanyang University, Korea, in 2003 and 2005 respectively. He is currently a senior member of engineering staff in ETRI, Korea. His research interests include Cyber-Physical System (CPS), Autonomic Computing and Software Engineering.



Ingeol Chun received his Ph.D., M.S. and B.S. degrees in Electrical and Computer Engineering from SungKyunKwan University, Korea, in 2011, 1998 and 1996 respectively. He is currently a senior member of engineering staff in ETRI, Korea from 1998. His research interests are Cyber-Physical Systems (CPS), Autonomic Computing, Agent-oriented intelligence system, Embedded systems and Software Engineering.



Wontae Kim received his B.E., M.E., and Ph.D. degrees in Electronic Engineering from Hanyang University, Seoul, Korea in 1994, 1996, and 2000, respectively. From Jan. 2001 to Feb. 2005, he was CTO of Rostic Technologies, a venture company which developed advanced mobile technologies. He joined ETRI (Electronics and Telecommunication Research Institute), the major national research institute of Korea, in March 2005. He is the team director of CPS (Cyber-Physical Systems) Research Team in Dept. of Embedded SW from Aug. 2010. His main research areas are CPS, RT Middleware, Autonomic Control, and High confidential Computing.