Contemporary Engineering Sciences, Vol. 9, 2016, no. 18, 853 - 862
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2016.6697

An Approach to Applying Goal Model and Fault Tree for Autonomic Control

Dongbeom Ko, Teayoung Kim and Jeongmin Park*

Dept. of Computer Engineering, Korea Polytechnic University
237 Sangideahak-ro, Siheung-si, Gyeonggi-do, South Korea
*Corresponding author

Sungjoo Kang and Ingeol Chun

CPS Research Lab, Electronic & Telecommunication Research Institute (ETRI)
218, Gajeong-ro, Yuseong-gu, Deajeon, South Korea

Abstract

In this study, autonomic control system using goal-model and fault-tree is proposed. Currently, the fourth industrial revolution is ongoing and manufacturing industrial project has been promoted in 4M1E (Man, Material, Machine, Method, Energy) environment and CPS (Cyber-Physical System), such as industry4.0 and smart-factory. The construction of intelligent manufacturing systems requires autonomic control system. However, existing research for the autonomic control system does not suggest a complete process (monitoring, analysis, planning strategy, executive strategy). Therefore, this study proposes an autonomic control system with a four step methodology that includes, 1) Goal-model constraints mapping, 2) Monitoring the goal-model, 3) Request problem recognition and diagnosis, 4) Problem diagnosis and strategy execution. This methodology can provide autonomic control system as entire process and is applicable to the real systems such as smart-factory.

Keywords: Goal-modeling, Autonomic Computing, Fault-tree

1 Introduction

Recently, manufacturing industry project has been promoted because of the emergent fourth industrial revolution [1] [2]. Autonomic control system has become a very important issue for construction of intelligent manufacturing systems. Existing autonomic control methodologies have advantage of goal modeling and fault detecting by abstracting the massive goals. Nevertheless, goal-model supports only monitoring & analysis section in autonomic system and fault-tree supports planning & executing strategy. Therefore, the present study proposes autonomic control system comprising 4 steps using goal-model and fault-tree, so as to solve this problem. For evaluation of autonomic control system, the proposed method was applied based on smart-factory environment and using fault-injection [3]. The current paper consists of various section, i.e., Section2 describes intelligent manufacturing system and autonomic control system methodology; Section3 introduces our methodology; Section4 explains experiment and evaluation; and Section 5 describes conclusion.

2 Background

This section describes Smart-factory that provides a backdrop for Inderstry4.0 and 4M1E environment, as well as analyses feature, strength, and weakness of the various autonomic control methodologies.

2.1 Autonomic control technology

The Fraunhofer Institute conducts research incorporating the ubiquitous technology in the manufacturing environment within the concept of Smart-factory. A typical example is the industrial automation solutions of Siemens. This solution is pre-simulation using CPS. Siemens's factory in Amberg, Germany, is evaluated by homologated European best plants with coupling of high productivity manufacturing and ICT techniques. In Korean, however, generically applied case and research of these technologies in the manufacturing system do not yet exist. Therefore, there is a requirement for a method that can reduce the energy and cost through autonomic control equipment in the plant.

2.2 4M1E environment [4]

4M management is the basic management element for improving the measurement and the working efficiency of the capacity utilization in the manufacturing industry. Element of 4M1E includes the following. Man representing the ability of the participants; material representing the quality, features, and use of range of material; machine representing the equipment of the manufacturing process; and method representing the technical solution, representative process and test method, and Energy. 4M1E facilitates efficient production, to eliminate waste, which could result in increased productivity of administrators and worker together with profit to target companies. Our study emphasizes all elements of the 4M excluding the Material. Among these, this study focuses on dealing with the Machine.

2.3 Goal-model based autonomic control methodology

Goal-based autonomic control methodology [5] allows abstracting a huge target, and has an advantage of easy design.

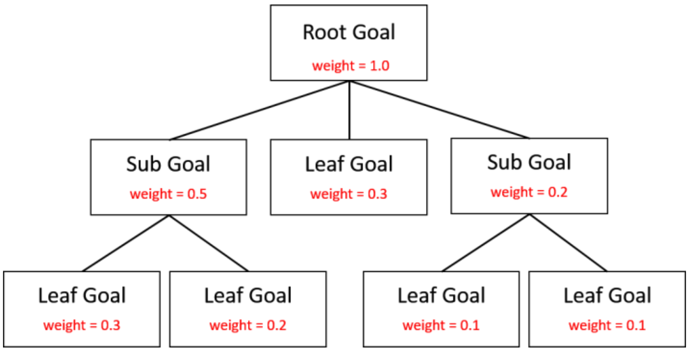

Fig 1. Goal-model

Autonomic control methodology, typically KAOS [6] and i*[7] exists. Goal-model methodology proposed to South Korea Information and Communication organization standard (TTAK.KO-11.0098)[8], improves the management skills of the developers; in addition, it is a methodology for the expected effect of reduction of administrative costs of the system. It has the advantage of easy modeling and planning strategy because it abstracts problem based on goal. In addition, it can detect the problem through goal-model and obtain overall goal achievement. However, goal-modeling needs additional methodology of resolution and reconstruction step because it provides the only stage of monitoring and analysis at the stage of autonomic computing.

2.4 Fault-tree based autonomic control methodology

Fault-tree based autonomic control methodology is to analyze the situation based on the fault-tree detection of an error.

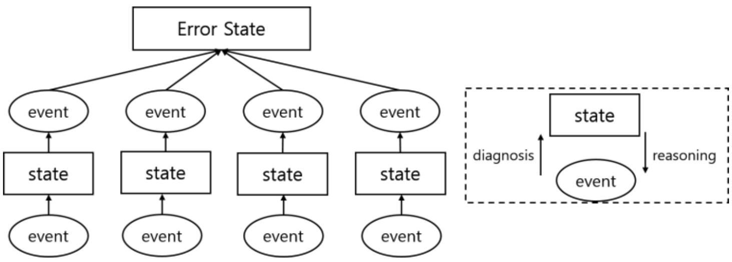

Fig 2. Fault-tree

It has a benefit that it is possible to adapt the strategy based on fault-tree. However, it is a difficult process that determines the planning strategy and fault analysis of a huge system, because the administrator determines strategy based on experience. These problems are solved by modeling the system's goal.

2.5 A common problem

Related studies do not suggest a complete autonomic control process i.e., monitoring, analysis, planning & execute strategy and difficult application to real autonomic control system and testing. Goal-model provides only monitoring and analysis step and is a very difficult process to decide strategy without abstracting the goal. Furthermore, there is a need to develop an autonomic control system for the recently emerged projects such as Industry4.0 and Smart-factory. Thus, our paper proposes autonomic control process complement disadvantage of existing autonomic control methodologies.

3 Autonomic control system processes and structures

This section explains the autonomic control process and architecture connection the goal-model and fault-tree.

3.1 Process of autonomic control system

Fig.3 shows process of proposed system that follows four steps. This section explains the autonomic control system.

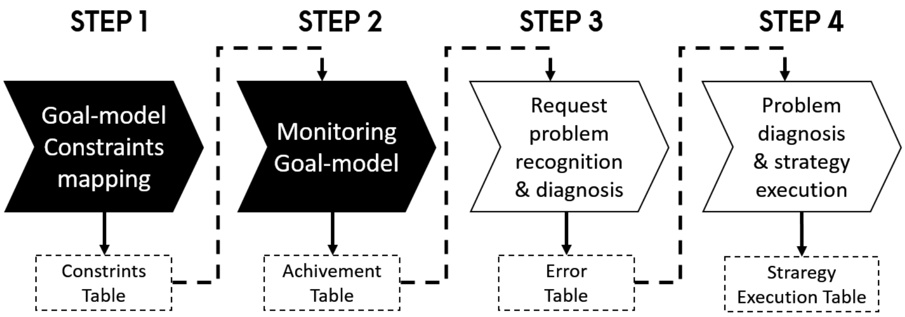

Fig 3. Autonomic Control Processes

Step1: Goal-model constraints mapping

Fig.1 describes goal-model. Goal-model consists of root-goal, sub-goal, and leaf-goal. Leaf-goal is linked to the constraint, which is an evidence for detecting whether or not the ordinary goal and fault-tree are related to goal's error (hexagon). Step1 compares constraint's value and measured value by sensor for detecting goal's achievement (see Fig.4).

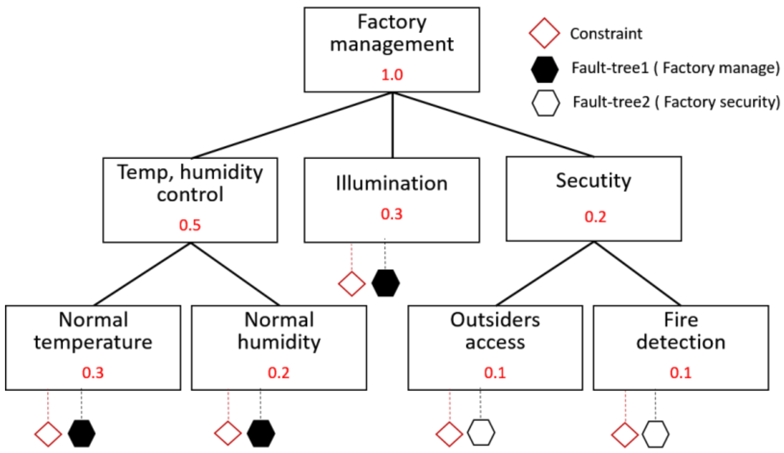

Fig 4. Goal-model for Smart-Factory

Step2: Monitoring goal-model

In the second step, determine the target achievement based on achievement of goals in Step1. Total achievement rate is 1, which is the same as root-goal's achievement. If goal dissatisfies constraint, which becomes dissatisfaction state, parent node containing the root-goal will have all weights reduced. Algorithm1 shows measurement of goal achievement and Algorithm2 presents change achievement rate when fault is detected.

Algorithm 1 Measurement Algorithm of Goal Achievement

Input:
$G[N] : goals$ ▷ N: number of goals
$w[N] : weights$
$weight \leftarrow 0$
for $i \leftarrow 1, N$ **do**
  if $G[i]$ $is\ leafgoal$ **then**
    $Stat \leftarrow Conf(G[i])$ ▷ Conf: function of confirm constraint
    if $Stat == true$ **then**
      $weight = weight + w[i]$
return $weight$

Algorithm 2 Change achivement rate when fault detect

Input:
$AG : abnomal\ goal$
$AG.state \leftarrow false$
$G \leftarrow AG.parent$ ▷ G is goal node
while $G.parent \neq Null$ **do** ▷ Decrease weight to root-goal
  $G.weight \leftarrow G.weight - AG.weight$
  $G \leftarrow G.parent$
$Adaptation = true$
return $Adaptation$

Step3: Request problem recognition and diagnosis

When the system calculates that the goal achievement is less than expected, autonomic control system will request problem recognition & diagnosis with fault-tree that connected goal be achieved.

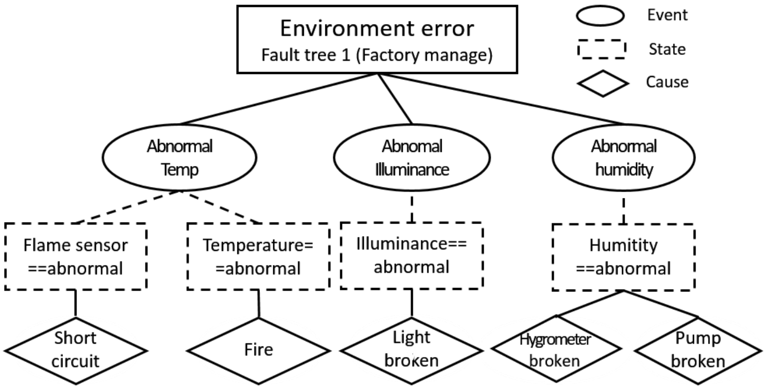

Fig 5. Fault-tree interprets fault state and decides strategy

Hexagon in the Fig.4 presents fault-tree id, which is related when goal's state is unsatisfied. One of the overall goals may have a number of fault-trees.

Step4: Problem diagnosis and strategy execution

Step4 involves diagnosis problem and executive strategy. Fault-tree interprets fault state and decides strategy by delivered information. Executing the strategy follows when strategy is determined.

3.2 Autonomic control system architecture

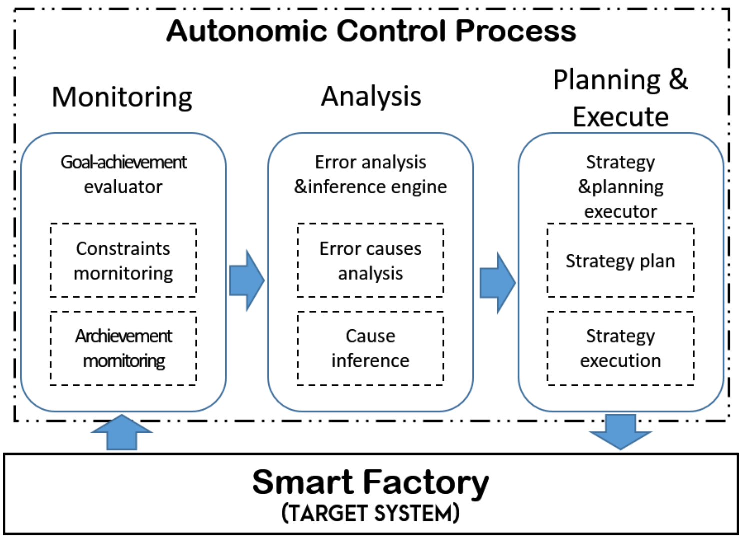

Fig 6. Autonomic control architecture

Structure of autonomic control system consists of Goal-achievement evaluator that monitors constraint and goal achievement, Error analysis & inference engine that conducts fault analysis and causes inference, and Strategy & planning executor that plans and executes strategy (see Fig.6). Fig.6 presents autonomic control system in Smart-factory.

Goal-achievement evaluator

Goal achievement evaluator monitors overall goal achievement through constraint, which is connected to goal. If it detects dissatisfaction goal, this information is delivered to Error analysis & inference engine.

Error analysis & inference engine

Error analysis & inference engine analyses fault and inference cause by using error analysis & inference table. Fig.5 shows the context. This decides adaptation strategy for solving problem and delivers strategy to Strategy & planning executor.

Strategy & planning executor

Strategy and planning executor determines adaptive strategies on the basis of the message received form the Error analysis & inference and executes strategy based on strategy execution table.

4 Experiment and evaluation

This section uses the fault-injection to the virtual Smart-factory system for experimentation and evaluation. Fig.7 describes fault-injection steps when scenario is ‘illumination abnormal’.

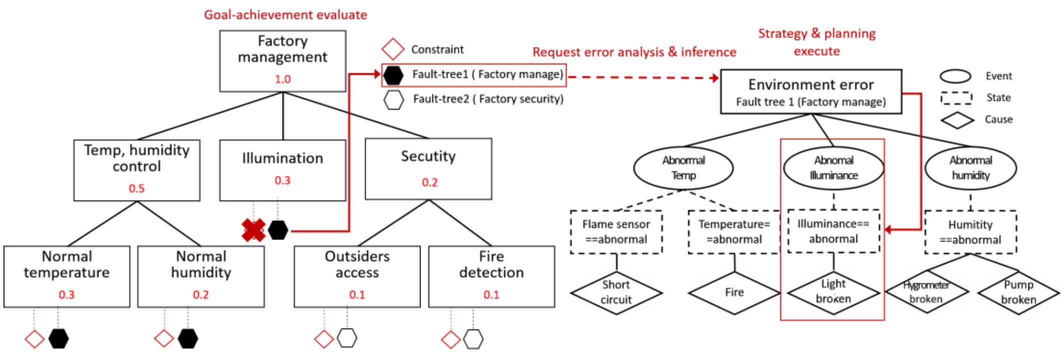

Fig 7. Experiment Scenario

Fault injection: ‘light sensor abnormal’

Injecting fault of ‘illumination = abnormal’ in the Smart-factory system, which is usually done. Therefore, system detects problem and Goal achievement evaluator, Error analysis & inference engine, and Strategy & planning executor control over their respective roles. 1) Goal achievement evaluator compares measured value that illumination sensor detected and constraint's value. If violated, goal 'illumination' becomes abnormal state. Goal achievement evaluator delivers constraint data that connected 'illumination' to Error analysis & inference engine. 2) Error analysis & inference engine analysis 'abnormal illuminate' state and inference that light was broken. 3) Strategy & planning executor executes strategy, which is connected cause of broken light.

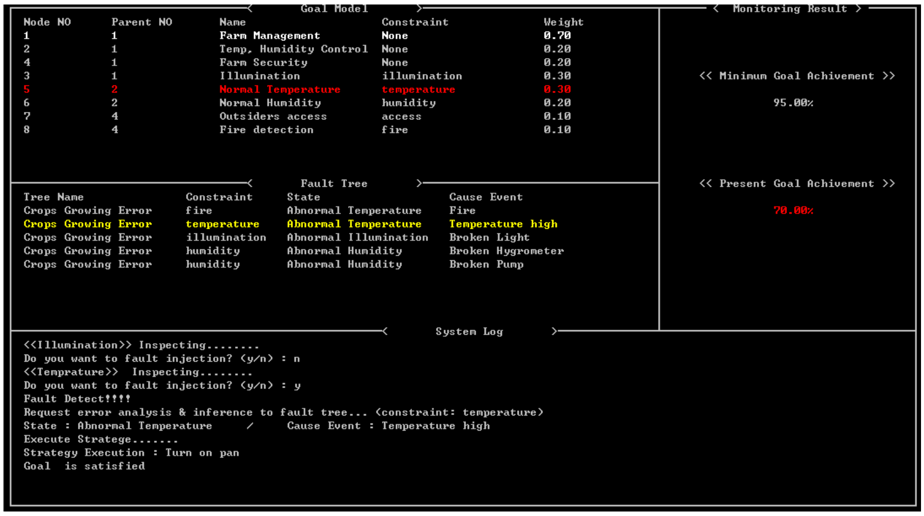

Fig 8. Simulation UI

Fig.8 shows simulation program user interface. Right side presents overall goal achievement rate, upper panel shows goal-model, fault-tree, and lower panel shows system log.

Evaluation

Our study evaluated fault detection rate over 5 times. Evaluation was divided into the fault-tree detection and goal-model detection for accurate assessment. Injection was the cause of errors in Fig.5 and the number of fault is increased by one for each. Fig.9 is graph of fault detection in the goal-model and fault-tree.

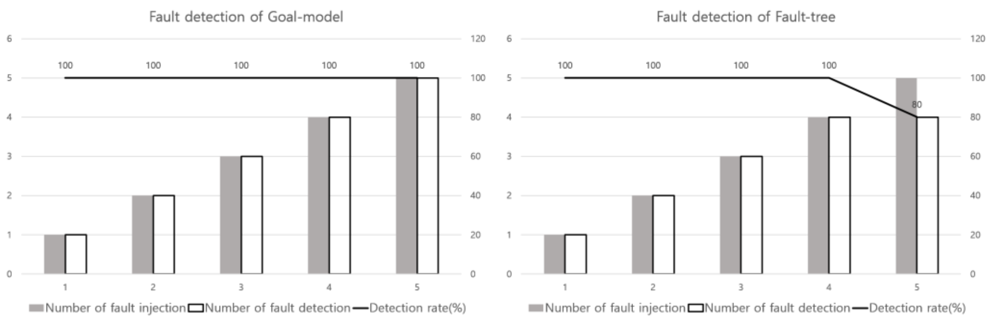

Fig 9. Graph of Fault detection

In fault detection of Goal-model, number of fault injection detection is same; but Fault detection of Fault-tree is not, since there are errors that have multiple causes. In Fig.5, for example, causes of 'humidity abnormal' are hygrometer broken and pump broken. However, this error detection constraint is 'Humidity=abnormal'. Thus, 'pump broken' error should be detected as 'hygrometer broken'. This can be answered by improving the fault-tree **[9]**.

5 Conclusions

This study proposed autonomic control system using goal-model and fault-tree. Through this suggestion, we designed complete autonomic control system by not fully integrated system methodologies. Nevertheless, the strategy when an unexpected fault occurs and fixed achievement need improvement. This can be solved by planning strategy dynamically and controlling goal's weight or defining standard achievement. In addition, this study requires network-based system not stand-alone for real Smart-factory system. Therefore, this study will be improved by 'machine learning' in autonomic system for dynamic decisions and network-based system in 4M1E environment [10] [11].

Acknowledgements. This work was supported by the ICT & R&D program of MSIP/IITP. [R-20150505-000691, IoT-based CPS platform technology for the integration of virtual-real manufacturing facility]

References

[1] Paulo Leitao, Agent-based distributed manufacturing control: A state-of-the-art survey, *Engineering Applications of Artifical Intelligence*, **22** (2009), no. 7, 979-991. http://dx.doi.org/10.1016/j.engappai.2008.09.005

[2] Weiming Shen, Qi Hao, Hyun Joong Yoon, Douglas H. Norrie, Applications of agent-based systems in intelligent manufacturing: An updated review, *Advanced Engineering Informatics*, **20** (2006), 415-431. http://dx.doi.org/10.1016/j.aei.2006.05.004

[3] L. Cheung, Early Prediction of Software Component Reliability, *Proceedings of the 13th International Conference on Software Engineering ICSE'08*, (2008), 111-120. http://dx.doi.org/10.1145/1368088.1368104

[4] Mao Yihua, Xu Tuo, Research of 4M1E's effect on engineering quality based on structural equation model, *Systems Engineering Procedia*, **1** (2011), 213-220. http://dx.doi.org/10.1016/j.sepro.2011.08.034

[5] J.M. Park, S.J. Kang, L.G. Chun, W.T. Kim, An Approach to Generationg Goal Model for Cyber-Physical Systems, *Proceeding of International Symposium on Embedded Technology*, **6** (2013), 347-357.

[6] Axel Van Lamsweerde and Emmanuel Letier, From object orientation to goal orientation: A paradigm shift for requirements engineering, Chapter in *Radical Innovations of Software and Systems Engineering in the Future*, Springer Berlin Heidelberg, 2004, 325-340. http://dx.doi.org/10.1007/978-3-540-24626-8_23

[7] Eric S. Yu, Social modeling and i*, Chapter in *Conceptual Modeling: Foundations and Applications*, Springer Berlin Heidelberg 2009, 99-121. http://dx.doi.org/10.1007/978-3-642-02463-4_7

[8] Requirements of Goal Modeling for Cyber-Physical Systems, Telecommunications Technology Associatoin (Korea), (2012).

[9] DongBeom Ko, Teayoung Kim, Sungjoo Kang, Ingeol Chun, Jeongmin Park, Research on Autonomic control System Conection Goal-model and Fault-tree, *Advanced Science and Technology Letters*, **129** (2016), 47-53. http://dx.doi.org/10.14257/astl.2016.129.10

[10] Yuyang Liu, Wooi Ping Cheah, Byung-Ki Kim and Hyukro Park, Predict Software Failure-prone by Learning Bayesian Network, *International Journal of Advanced Science and Technology*, **1** (2008), 35-42.

[11] Anestis A. Toptsis, Reflective Thinking, Machine Learning, and User Authentication via Artificial K-lines, *International Journal of Advanced Science and Technology*, **5** (2009), 51-74.

Received: May 1, 2016; Published: August 11, 2016