

Using Qualitative Reasoning in Building Ubiquitous Computing System

Hyeon-Kyeong Kim

Department of Information Science and Telecommunication, Hanshin University
411 Yangsan-dong, Osan-si, Gyeonggi-do, 447-791, Korea
hkim@hs.ac.kr

Abstract

A key problem in building ubiquitous computing system is providing context aware and adaptive services to users by cooperation among various computing objects. In this paper, we show how an integration of qualitative reasoning and multi-agent based architecture can be used to provide context aware and adaptive services for operation of physical system in ubiquitous computing environment. Differential qualitative analysis is used to propose appropriate services once an abnormal operation of a given physical system is detected. Causal dependencies generated from a qualitative model combined with primitive tasks are traced to restore the given physical system to normal state. This idea has been implemented and tested on several tasks including the operations of incineration plant and pyrolysis reactor.

Introduction

Recently, computing-device-rich and communication-resource-rich environment has introduced a new computing paradigm called ubiquitous computing. The goal of ubiquitous computing is to provide seamless services to users by cooperation between various computing objects. A key factor in building ubiquitous computing system (UCS) is intelligence that can provide self-configuring, self-repairing and adaptive abilities (Roman et al, 2002; Ranganathan, 2005). We believe that qualitative reasoning (QR) will play an important role in providing the intelligence in UCS. We expect that QR can provide a way for breadth, robustness and natural user interaction. Since QR has been developed to capture both everyday reasoning and expert reasoning, it can provide a firm ground for breadth of system. By robustness, we mean the ability to solve a given problem even with partial knowledge. QR, that solves problems by using qualitative knowledge, already provides a good possibility (Klenk et al, 2005). Lastly, natural interaction with user is an important feature for UCS since it can enhance active user participations as active computing objects in cooperation. Qualitative nature of QR is expected to support this feature.

We are exploring how QR can support intelligent services for a given task in ubiquitous computing

environment. Unlike previous work in task execution framework in ubiquitous computing that workflow and primitive tasks are predefined, our framework accomplishes a given task by tracing causal dependencies generated for the task (Ranganathan, 2005). We are focusing on the operation of physical system. In this paper, we show our work by describing a UCS for operation of incineration plant (Kim & Kim, 2007).

Incineration Process Domain

Lately, incineration has drawn attention because of its reduction efficiency of solid wastes and technical stability. A lot of incineration plants have been installed in Korea during the last two decades. However, incineration was blamed to generate air pollutants such as dioxins. Basically it consists of 1st incinerator, 2nd incinerator and flue gas treatment train. Given solid waste and air, it finally produces gases through combustion and thermal destruction of incompletely combusted gas. Ideally if the working condition is controlled, this process should not produce air pollutants. But it is impossible in real world. What we can do is to control parameters of system to avoid undesired conditions as much as possible.

While many efforts have been made to develop complete quantitative incineration process model, complicated thermodynamic nature of the process has made it difficult to build such a model. Although such a model has been developed, there are still difficulties to get complete and precise input data, i.e., complete data about solid waste supplied. In addition to this, it is not easy for non-expert to interpret numerical results.

Problem Solving with a Qualitative Model

A qualitative model provides a basis for problem solving. The expert knowledge about incineration process is often imprecise and of qualitative nature. Our qualitative incineration process model captures the qualitative knowledge of the experts. It is built based on qualitative

process theory (Forbus, 1984). Figure 1 shows a part of causal dependencies generated from a qualitative incineration model.

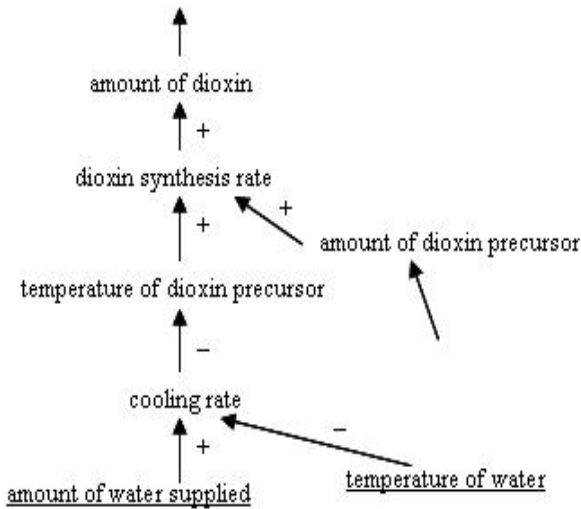


Figure 1: A part of causal dependencies of incineration process

We are focusing on the operation of physical system. Our system concentrates on maintaining normal conditions. Once an abnormal condition occurs, our system tries to restore to a normal state. It implies that system tries to the related quantity to change (e.g., to increase or to decrease) to a normal state. Differential qualitative analysis (DQA) is an inference technique that propagates the effects of a perturbation through causal dependencies (Weld, 1988). We use DQA backwards to find appropriate services for a given task. Problem solver works backwards from the top-level goal to sub-goals by tracing qualitative influences through the causal dependencies. For example, as shown as Figure 1 a way to decrease the amount of dioxin is to decrease dioxin synthesis rate. And one way to decrease the rate is to decrease the temperature of dioxin precursor. And the temperature is decreased by increasing cooling rate. The cooling rate is increased by increasing the amount of water supplied. The rate is also increased by decreasing the temperature of water. Then the task to increase the cooling rate is achieved by primitive tasks to control water supplier device and water temperature control device, respectively.

In the figure, underlined parameters represent primitive tasks that are controlled by operating corresponding devices. Services in our framework are primitive tasks performed by devices. To achieve a task, reasoning agent regress back to the primitive tasks through causal relations.

UCS for Operation of Physical System

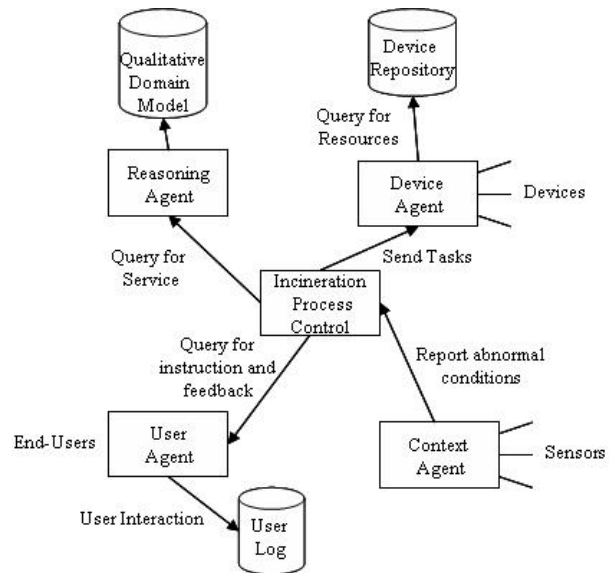


Figure 2: Architecture

Figure 2 shows the overall architecture for our system. In this section, we describe our system by using operation of incineration plant. The system (IPC: Incineration Process Control) consists of multiple agents that execute tasks by cooperation among them. We assume devices and sensors are connected to our framework. The following describes each agent shortly.

Context Agent: Context agent monitors the chosen plant. It collects data about quantities from the sensors attached to the plant, detects deviations from expected values of quantities: either a higher value than expected or a lower value than expected. Once a deviation is detected, this fact is reported to IPC. For example, if the concentration of dioxin is higher than normal value, this is reported to IPC.

Reasoning Agent: Reasoning agent finds solutions, i.e., proper services to change higher or lower value to expected value. It tries to make the quantity with a higher value decrease and to make the quantity with a lower value increase. It figures out the necessary service by applying DQA through causal dependencies generated from qualitative model. Service consists of primitive tasks operated by devices such as “increase water supply by operating water supplier”.

Device Agent: Once proper service is suggested by reasoning agent, IPC requests to device agent for relevant devices to perform primitive tasks as directed. Device repository maintains a list of all entities such as incineration plants and their devices. Devices are the entities that can change the values of physical parameters such as air supplier and thermostat.

User Agent: User agent provide user interface. The explanation regarding the execution of the task is given to users and users provide feedback through user agent.

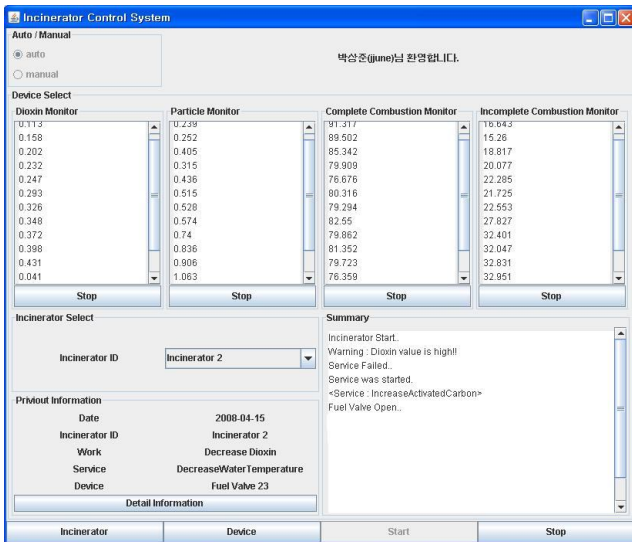


Figure 3: Screenshot of user interface

Figure 3 shows screenshot of user interface in the middle of execution of IPC. The top four windows show the values of quantities monitored by corresponding sensors. Since dioxin level is reported as higher than expected in the figure, reasoning agent finds proper services to lower the level. The window at the right bottom displays explanation regarding the perturbation of the quantity and the execution of the services performed by device agent.

Our system has been implemented in Java. Qualitative incineration model and reasoning agent have been implemented by using Jess. Jess is a rule engine for the Java platform. Agent communications has been implemented with JADE. JADE is Java agent Development Framework that supports developing multi-agent system.

Discussion

This work shows our effort to combine QR with multi-agent framework for building an intelligent UCS. We are exploring how QR can provide context aware and adaptive services in ubiquitous computing environment. We believe that UCS is a good application domain where QR can show its strengths. We plan to keep exploring the possibilities.

Acknowledgement

This research is supported by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy(MKE) 21st Century

Frontier R&D Program in Korea and a result of subproject UCN 08B3-S2-10M.

References

- Forbus, K. 1984. *Qualitative Process Theory*. Artificial Intelligence
 Jess. <http://herzberg.ca.sandia.gov/jess/>
 JADE. <http://jade.tilab.com/>
 Klenk, M. et al. 2005. *Solving Everyday Physical Reasoning Problems by Analogy using Sketches*. AAAI
 Kim, H. and Kim, S. 2007. *Multi-agent Based Incineration Process Control System with Qualitative Model*. 10th Pacific Rim Int'l Workshop on Multi-Agents
 Ranganathan, A.. 2005. *A Task Execution Framework for Autonomic Ubiquitous Computing*. PhD thesis, University of Illinois at Urbana-Champaign
 Roman, M. et.al, 2002. *Gaia: A middleware Infrastructure to Enable Active Spaces*. IEEE Pervasive Computing
 Weld, D. 1988. *Comparative Analysis*. Artificial Intelligence.