International Journal of New Computer Architectures and their Applications (IJNCAA) 4(1): 48-55
The Society of Digital Information and Wireless Communications, 2014 (ISSN: 2220-9085)

# Developing Core Software Requirements of Energy Management System for Smart Campus with Advanced Software Engineering

Shigeru KUSAKABE, Hsin-Hung LIN, Yoichi OMORI, and Keijiro ARAKI
Graduate School of Information Science and Electrical Engineering, Kyushu University
744 Motooka Fukuoka, 819-0395, Japan
E-mail: kusakabe@ait.kyushu-u.ac.jp

## ABSTRACT

Higher educational institutions such as universities provide campus data and information to stakeholders such as students, faculty staff, administrators, media, and community members. They also analyze such data and information for decision making, planning and continuous improvement. They obtain necessary capabilities for collecting, providing and analyzing data and information by acquiring products and services from suppliers or by developing products and services by themselves. As in today's social infrastructure, software plays an important role and software seems to keep growing in its scope, size and complexity in infrastructure of educational institutions. However, many organizations have not invested in the capabilities necessary to effectively manage life-cycle process for such software products and services. We argue that software engineering techniques, such as process improvement framework and software product line engineering are important in developing requirements and systems effective in continuous improvement. In this paper, we discuss a trial case to improve the energy management system in our campus, as an emerging area of institutional continuous improvement, by using software engineering techniques under the goal of increasing the awareness and involvement of members for smart energy management.

## KEYWORDS

Energy management system, Smart campus, Institutional research, Software development process, Software product line engineering, Formal methods

## 1 INTRODUCTION

Higher educational institutions such as universities and colleges provide campus data and information to stakeholders such as students, faculty, administrators, media, and community members. Many educational institutes have their institutional research office to supports collection of evidence, data, and information to conduct analyses for decision making, planning and continuous improvement in the institutes. Institutional research offices cover wide range of areas such as admissions, student services, career outcomes, and facilities, and seem to become covering wider areas including emerging areas such as sustainability[1].

They obtain necessary capabilities for collecting, providing and analyzing data and information by acquiring products and services from suppliers or by developing such products and services by themselves. As in today's social infrastructure, software plays an important role and software seems to keep growing in its scope, size and complexity in infrastructure of educational institutions. However, many organizations have not invested in the capabilities necessary to effectively manage life-cycle process for such software products and services. We argue that advanced software engineering paradigms, such as process improvement framework[2] and software product line engineering[3] are important in developing requirements and systems effective in continuous improvement.

In this paper, we discuss a trial case to improve the energy management system in our campus. As an emerging area, sustainability has been becoming one of the important issues in educational institutes, and sustainability has been becoming an area to be covered in continuous improvement in educational institutes. Many universities are trying to make their campuses sustainable, and one of the aspects of sustainability is smart energy management. We used software engineering techniques such as software product line engineering under the goal of increasing the awareness and involvement of members for smart energy management.

The rest of this paper is organized as follows. We firstly introduce a few key paradigms in software engineering useful in continuous improvement in higher educational institutes. In Section 2, we briefly introduce a process model framework, the CMMI process improvement model framework, more specifically, focusing on process areas related to requirements development. In Section 3, we discuss requirements development for a unit suitable for a university campus under the goal of increasing the awareness and involvement of members for smart energy management. We briefly introduce product line software engineering in analyzing various energy management systems and developing requirements for smart energy management in campus. In Section 4, we evaluate our approach through prototyping with our requirement as requirement validation. Section 5 concludes this paper.

## 2 PROCESS MODEL FRAMEWORK

In this section, we briefly introduce a process model framework, CMMI process improvement framework developed by the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU). More specifically, we focus on process areas related to requirements development, as one of the major problems in acquiring the energy management systems is a problem of requirements due to the following factors.

- There is a variation in purposes and subjects among different energy management systems.
- Requirements will continuously change in a fast pace according to the changes in the social circumstance and technology innovation. This applies to different phases of the life-cycle, such as operation and maintenance as well as acquisition,

### 2.1 CMMI Process Models

Organizations obtain necessary capabilities for continuous improvement by acquiring products and services from suppliers or by developing such products and services by themselves. However, many organizations have not invested in the capabilities necessary to effectively manage lifecycle process for software-based products and services. The CMMI models are collections of best practices that help organizations to improve their processes in acquiring and managing necessary capabilities. They embody the premise “the quality of a system or product is highly influenced by the quality of the process used to develop and maintain it”. Best practices in the models are collected from government and industry. There exist specific models in CMMI[4], and each of CMMI models is generated from the CMMI Architecture and Framework:

- CMMI for Development (CMMI-DEV): Product and service development,
- CMMI for Services (CMMI-SVC): Service establishment, management, and
- CMMI for Acquisition (CMMI-ACQ): Product and service acquisition.

The CMMI models include process areas, each of which is a cluster of related practices in the area of interest. We can satisfy a set of goals considered important for making improvement in the area when we implement the practices in the process area collectively. All CMMI models contain 16 core process areas, each of which covers basic concepts that are fundamental to process improvement in any model of interest. Model components are grouped into three categories - required, expected, and informative - that reflect how to interpret them. The required components are the specific and generic goals that are essential to achieving process improvement in a given process area. The expected components are the specific and generic practices describing the activities that are important in achieving a required CMMI component.

The CMMI models provide two approaches to process improvement. These approaches are associated with two types of representation called continuous and staged. When using the continuous representation, organizations select an individual process area or a set of process areas and incrementally improve processes corresponding to the selected process area or the process area set. As we explain later, we focus on the process area for requirements development by using continuous representation. We briefly explain CMMI-ACQ[4] and CMMI-DEV[5] as organizations typically obtain necessary capabilities for continuous improvement by acquiring products and services from suppliers or by developing such products and services by themselves.

### 2.2 CMMI-ACQ

The CMMI-ACQ provides a comprehensive integrated set of guidelines for acquiring products and services. The CMMI-ACQ has 22 process areas presented in alphabetical order by acronym as follows:

- Agreement Management (AM)
- Acquisition Requirements Development (ARD)
- Acquisition Technical Management (ATM)
- Acquisition Validation (AVAL)
- Acquisition Verification (AVER)
- Causal Analysis and Resolution (CAR)
- Configuration Management (CM)
- Decision Analysis and Resolution (DAR)
- Integrated Project Management (IPM)
- Measurement and Analysis (MA)
- Organizational Process Definition (OPD)
- Organizational Process Focus (OPF)
- Organizational Performance Management (OPM)
- Organizational Process Performance (OPP)
- Organizational Training (OT)
- Project Monitoring and Control (PMC)
- Project Planning (PP)
- Process and Product Quality Assurance (PPQA)
- Quantitative Project Management (QPM)
- Requirements Management (REQM)
- Risk Management (RSKM)
- Solicitation and Supplier Agreement Development (SSAD)

As one of the major problems in acquiring the energy management systems is a problem of requirements, we briefly introduce the process area related to requirements development in the CMMI-ACQ, ARD, among the above-mentioned 22 process areas. The purpose of ARD is to elicit, develop, and analyze customer and contractual requirements. Specific goals and practices in this process area are summarized as follows:

- SG 1 Develop Customer Requirements
  - SP 1.1 Elicit Stakeholder Needs
  - SP 1.2 Develop and Prioritize Customer Requirements
- SG 2 Develop Contractual Requirements
  - SP 2.1 Establish Contractual Requirements
  - SP 2.2 Allocate Contractual Requirements
- SG 3 Analyze and Validate Requirements
  - SP 3.1 Establish Operational Concepts and Scenarios
  - SP 3.2 Analyze Requirements
  - SP 3.3 Analyze Requirements to Achieve Balance
  - SP 3.4 Validate Requirements

### 2.3 CMMI-DEV

The CMMI-DEV provides guidance for applying CMMI best practices in a development organization. The CMMIDEV has process areas for development in addition to the core process areas. The collection of best practices in the CMMI-DEV model focuses on activities for developing quality products and services to meet the needs of customers and end users. The CMMI-DEV has totally 22 process areas as listed in alphabetical order by acronym.

- Causal Analysis and Resolution (CAR)
- Configuration Management (CM)
- Decision Analysis and Resolution (DAR)
- Integrated Project Management (IPM)
- Measurement and Analysis (MA)
- Organizational Process Definition (OPD)
- Organizational Process Focus (OPF)
- Organizational Performance Management (OPM)
- Organizational Process Performance (OPP)
- Organizational Training (OT)
- Product Integration (PI)
- Project Monitoring and Control (PMC)
- Project Planning (PP)
- Process and Product Quality Assurance (PPQA)
- Quantitative Project Management (QPM)
- Requirements Development (RD)
- Requirements Management (REQM)
- Risk Management (RSKM)

- Supplier Agreement Management (SAM)
- Technical Solution (TS)
- Validation (VAL)
- Verification (VER)

Similar to the case of acquisition, one of the major problems in developing the energy management systems is a problem of requirements. The purpose of Requirements Development (RD) in the CMMI-DEV is to elicit, analyze, and establish customer, product, and product component requirements.
Specific goals and practices in the RD process area are summarized as follows:

- SG 1 Develop Customer Requirements
  - SP 1.1 Elicit Needs
  - SP 1.2 Transform Stakeholder Needs into Customer Requirements
- SG 2 Develop Product
  - SP 2.1 Establish Product and Product Component Requirements
  - SP 2.2 Allocate Product Component Requirements
  - SP 2.3 Identify Interface Requirements
- SG 3 Analyze and Validate Requirements
  - SP 3.1Establish Operational Concepts and Scenarios
  - SP 3.2Establish a Definition of Required Functionality and Quality Attributes
  - SP 3.3 Analyze Requirements
  - SP 3.4 Analyze Requirements to Achieve Balance
  - SP 3.5 Validate Requirements

In the rest of this paper, we assume an organization obtains necessary capabilities for continuous improvement by developing such products and services, and tries to improve the RD process area in the CMMI-DEV.

## 3 REQUIREMENTS DEVELOPMENT

Developing requirements is important in obtaining systems effective in continuous improvement for smart campus. In this section, we discuss a trial case to develop requirements to increase the awareness and involvement of stakeholders for smart energy management in our campus. We use software engineering methods, such as software product line engineering and formal methods useful in improving requirements development.

### 3.1 Energy Management in Campus

Many universities are trying to make their campuses sustainable and smart. As our government has been repeatedly requesting demand restraint every summer and winter in recent years, our university accordingly has been planning measures for energy saving as follows regarding hardware aspects:

1. Lighting equipment
   - Thinning of the light bulb and fluorescent lumps, while ensuring necessary illumination.
   - Consolidation of open lecture room
2. Experimental equipment
   - Consolidation, reduction of operating time, cooperative scheduling of laboratory equipment as much as possible
3. Office appliances
   - Consolidation of printer
   - Thinning of the available electric water heater
   - Consolidation of the refrigerator

Regarding operations and human related activities:

1. Lighting equipment
   - Turning off unnecessary lighting equipment
   - Extinguish lights during lunch break
2. Experimental equipment and office appliance
   - Reduction of standby power consumption
   - Enlightenment of power saving mode usage
3. Air-conditioning and other utilities
   - Limitation of upper (lower) room temperature
   - Limitation of operation period
   - Filter maintenance
   - Reduction of elevator usage (encourage of usage of stairs (yardstick: within 5 floors))
   - Keeping lid of the heating toilet seat closed and setting at the lowest temperature
4. Lifestyle
   - Enforcement of warm biz
   - Utilization of a rug, such as inner wear
   - Sending e-mails to employees when demand reduction is necessary

These measures are roughly divided into two points of view: a building manager's point of view and an individual resident's point of view. We expect energy management systems cover these points to promote energy saving activities of all the members in our campus. However, our current energy management systems are basically building management systems and covers only building manager's point of view. Reports from our energy management systems are summarized at a coarse grained level. It seems helpless to promote energy-saving awareness of the members at individual level. We need a proper energy management system effective in promoting sustainability for whole stakeholder in university campus.

Figure 1 shows overview of energy management systems of one of our campuses. Although we have energy management systems at building level, their outputs do not seem helpful in promoting our activities for sustainability. One of the main reasons is the gap between the unit of management and the unit to which we can make interaction. Figure 2 shows a sample chart generated by our building energy management system. Each vertical bar shows a total of energy consumption in a building and it is difficult to make a connection between the activities of each member and their consequence with the graph.

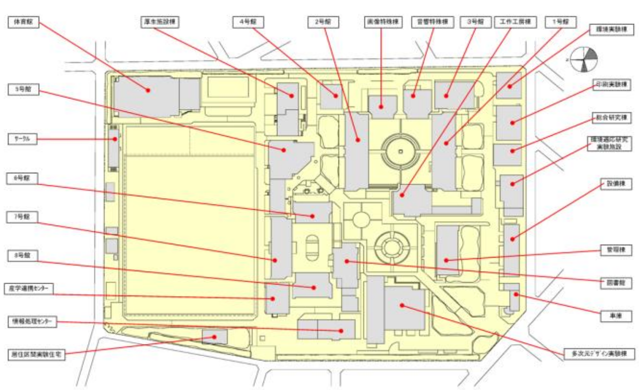

**Figure 1.** Overview of energy management systems of one of our campuses.

We assume a laboratory member is an important stakeholder, and we put emphasis on a view point of laboratory as one of the primary administrative units in university campus, where many of the students in certain grades and faculty members spend their time.

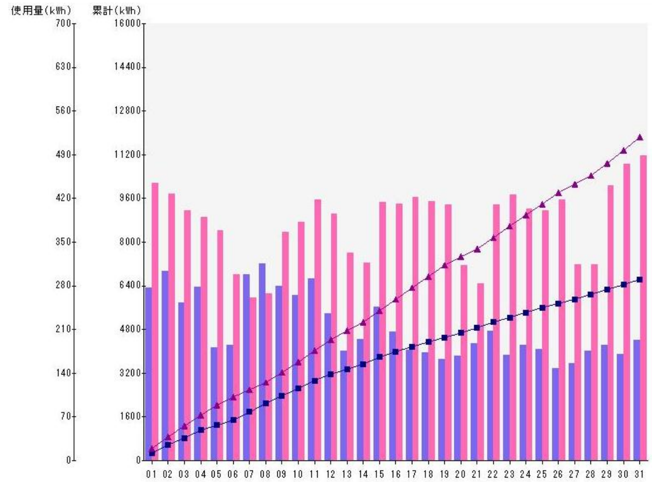


**Figure 2.** Sample chart generated by our building energy management system.

### 3.2 Software Product Line Engineering

In software product line engineering[3], most software systems are not regarded as completely new but belonging to a family of software systems with commonalities and variabilities. As shown in Figure 3, the software product line engineering paradigm has two processes, the domain engineering and the application engineering. The domain engineering process is responsible for establishing a reusable platform, a collection of reusable artifacts such as requirements and design in the software context. The application engineering process is responsible for deriving product line applications from the platform and ensuring the correct binding of the variability according to the specific needs of each application. This abstraction can be leveraged to improve the software system acquisition process as well as the software system development process.

Energy management systems suitable for university campus seem different from other energy management systems such as home energy management systems and building energy management systems, although they share a lot between them. We need to consider commonalities and variabilities among energy management systems, while energy management systems are becoming software intensive like many other infrastructure systems. Thus, software product line engineering paradigm seems to be becoming more important in developing energy management systems. We use domain engineering techniques in developing requirements of energy management systems for smart campus.

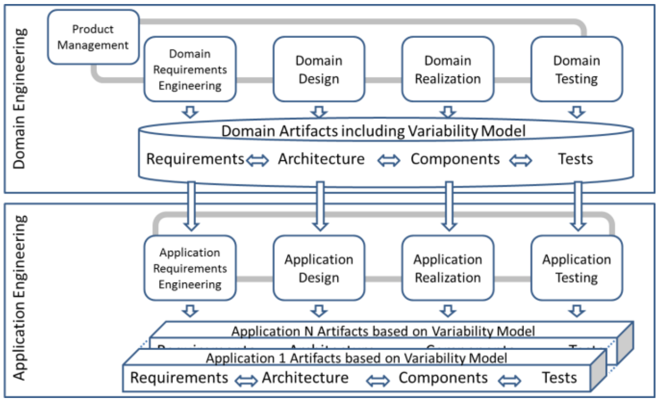


**Figure 3.** Outline of software product line engineering

We assume a laboratory is a suitable unit of energy management for smart campus as many of the students in certain grades and many faculty members spend most of their time in a laboratory. Most of administrative processes are conducted based on a unit of laboratory.

### 3.3 Analysis for Commonalities and Variabilities in Requirements

One of the major problems in developing the energy management systems is a problem of requirements due to the following factors.

- There is a variation in development purposes and subjects among different energy management systems.
- Requirements, in different phases of the lifecycle, such as operation and maintenance as well as development, will continuously change in a fast pace according to the changes in the social circumstance and technology innovation.

In addition to these issues of general energy management systems, there exit some issues specific to university campuses.

- Each campus has unique characteristics as well as common characteristics.
- As a target of energy management, we have a variation logically and physically.
- The management units impossible in the past can be realized thanks to technology innovations.
- Management policies may change due to social and financial conditions specific to university.

We develop requirements of energy management system for smart campus, taking into account of the issues described above.

There exist a variety of targets of energy management system, such as home energy management system (HEMS), building energy management system (BEMS), factory energy manage system (FEMS) and community energy management system (CEMS). Requirements for each target have commonalities and variabilities, and will be changed due to a variety of reasons such as social circumstance and technological innovation. In order to deal with these issues, the core requirements of the system are important. Using the techniques of domain analysis in software product line engineering, we analyze the energy management domain to determine the core requirements for laboratory energy management system in university campus.

From the view point of laboratory management, the policy of laboratory is affected by the policy of the building to which the laboratory belongs. At the same time, a laboratory accommodates a group of residents who have their own life style. They usually spend their daytime in their laboratory and may spend their nighttime, too. We analyze the requirements to cover assumed scenarios by examining following HEMS and BEMS products.

- Smart HEMS
- NEC Cloud-type HEMS
- Denso HEMS Monitor
- Navi-ene
- Panasonic ECO-SAS DR
- Enesys
- Enetune-BEMS
- Azbil BEMS

We identified commonalities such as publishing energy consumption data and allowing download of energy consumption data. We also identified variabilities in many points, such as the way of user-notification, methods to control the device, threshold user can specify, behavior in exceeding the threshold, target period of prediction, format of display, display device, target data type, period of measurement, target area, and conversion method.

## 4 EVALUATION THROUGH PROTOTYPING

In the specific practice SP 3.5, Validate Requirements, in the RD process area in the CMMI-DEV, examples of techniques used for requirements validation include the following: Analysis, Simulations, Prototyping, and Demonstrations. We chose Prototyping and developed a prototype to validate our requirements in our experimental framework where we have been studying the effectiveness of process improvement frameworks[7][8].
In order to deal with the variabilities as seen above in an incremental and manageable way, we use a team software development process, TSPi (Team Software Process introduction) [9]. In TSPi, we strategically develop the target product throughout the multiple development cycles as shown in Figure 4. Each cycle is divided into eight phases, such as launch/re-launch, strategy development, planning, requirements, design, implementation, testing, and postmortem. We start from the development for the basic requirements and restart the next development for the updated requirements after finishing one development cycle to in an incremental manner.

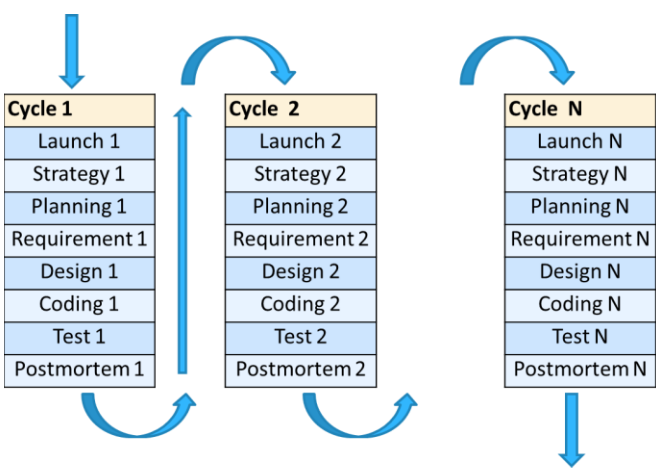


**Figure 4.** Outline of TSPi.

As a prototype, we developed a system whose sample output is shown in Figure 5. In this system, we use energy consumption data and floor plan for each laboratory. A member of the laboratory can see the energy consumption of the rooms used by the laboratory. This system seems more useful in increasing the awareness and involvement of laboratory members for smart energy management compared with coarser grained BEMS-based output like Figure 2.

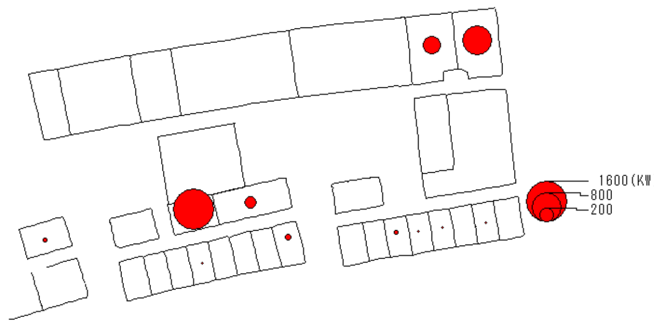


**Figure 5.** Sample output of a laboratory energy management system.

As an approach of using more advanced software engineering, we also examined the effectiveness of formal methods in requirements development in the domain engineering process in software product line engineering. Although we can use different approaches combining software product line engineering and formal methods[10] [11], we used a model-oriented formal method VDM[12] in this case study. By using VDM in developing and analyzing requirements in an efficient but disciplined way, we could find defects on data types frequently used in energy management systems at a requirements level before going into the design.

## 5 CONCLUDING REMARKS

We explained the capabilities necessary to effectively manage life-cycle process for software-based products and services are important for continuous improvement also in educational institutes. They use software-based systems to conduct analyses for decision making, planning and continuous improvement while collecting evidence, data, and information. In this paper, we explained process improvement framework and software product line engineering to develop requirements and systems effective in continuous improvement. We reported a trial case to improve the energy management system in our campus. We could develop effective requirements and a prototype more useful in increasing the awareness and involvement of members in the campus for smart energy management compared with conventional BEMS-based systems.

## 6 REFERENCES

[1] Institutional Sustainability Plan - University of Calgary, http://www.ucalgary.ca/sustainability/files/sustainability/InstitutionalSustainabilityPlan.pdf

[2] Process and Performance Improvement, http://www.sei.cmu.edu/process/index.cfm

[3] K. Pohl, G. Bockle, and F. Van Der Linden. Software Product Line Engineering: Foundations, Principles, and Techniques. Springer, Heidelberg, 2005.

[4] CMMI solutions. http://cmmiinstitute.com/cmmi-solutions/

[5] CMMI for Acquisition. http://www.sei.cmu.edu/reports/10tr032.pdf

[6] CMMI for Devlopment. http://www.sei.cmu.edu/reports/10tr033.pdf

[7] S. Kusakabe, Y. Omori and K. Araki, A Combination of a Formal Method and PSP for Improving Software Process, TSP Symposium 2012.

[8] Shouhei Kanamaru, et al. Case Report of Collaboration PBL between Industry and Academia Based on TSPi - Introduction of Formal Methods to OpenFlow Software-, IPSJ SIG Technical Report, Vol.2013-SE-179-26, pp.1-8, 2013. (In Japanese).

[9] Humphrey, Watts S: Introduction to the Team Software Process(sm), Addison-Wesley Professional, 2000.

[10] T. Kishi and N. Noda. Formal verification and software product lines. Communications of the ACM, 49(12):73–77, 2006.

[11] Dave Clarke, Nikolay Diakov, Reiner Hähnle, Einar Broch Johnsen, Germán Puebla, Balthasar Weitzel, and Peter Y. H. Wong. HATS - A Formal Software Product Line Engineering Methodology. SPLC Workshops, page 121-128. Lancaster University, (2010)

[12] VDM information, http://www.vdmtools.jp/en/