Requirements Development of Energy Management System for a Unit in Smart Campus

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
kusakabe@ait.kyushu-u.ac.jp

***Abstract*—Institutional research supports collection of evidence, data, and information to conduct analyses for decision making, planning and continuous improvement in educational institutes. Conceptually, it covers wide range of areas such as admissions, student services, career outcomes, and facilities. It seems to become covering wider areas including emerging areas such as sustainability. As in today's social infrastructure, software plays an important role also in institutional domain. Software seems to keep growing in its scope, size and complexity also in institutional research domain. However, many organizations have not invested in the capabilities necessary to effectively manage life-cycle process for such products and services. We argue that software engineering techniques, such as life-cycle process model and software product line engineering are important in developing systems effective in institutional research. In this paper, we discuss a trial case of requirements development in improving the energy management system in our campus, as an emerging area of institutional research, 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, Software requirements development, Software product line engineering**

I. Introduction

Institutional research supports collection of evidence, data, and information to conduct analyses for decision making, planning and continuous improvement in educational institutes. Conceptually, it covers wide range of areas such as admissions, student services, career outcomes, and facilities. It seems to become covering wider areas including emerging areas such as sustainability in order to reflect the changes of the society and stakeholder interests.

Actual institutional research typically uses systems based on information technologies, and organizations acquire necessary capabilities for institutional research by obtaining products and services from suppliers or by developing such products and services within themselves. As in today's social infrastructure, software plays an important role in institutional domain and software seems to keep growing in its scope, size and complexity also in institutional research domain. However, many organizations have not invested in the capabilities necessary to effectively manage life-cycle process for software-based products and services from the view point of institutional research. We argue that advanced software engineering technologies, such as process improvement models[1] and software product line engineering[2] are important in developing systems effective in institutional research systems.

In this paper, we discuss a trial case of requirement development in improving the energy management system in our campus. Many universities are trying to make their campuses sustainable, and one of the aspects of sustainability is smart energy management. As an emerging area, sustainability has been becoming an area to be covered by institutional research[3]. We used software engineering techniques 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 introduce a few key technologies in software engineering also useful in institutional research. In Section II, we briefly introduce a process model framework, the CMMI process improvement model framework, specifically focusing on the requirements development process area. In Section III, 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 IV, we evaluate our approach through prototyping with as requirement validation. Section V concludes this paper.

II. Process Improving Model

There exist process improvement frameworks in the literature to help organizations optimize their underlying processes to achieve their goals. One of the important aspects of institutional research is data management. There exist maturity models for data management such as DMBOK (Data Management Book of Knowledge) from DAMA (Data Management Association) [4] and DMMI (Data Management Council Data Management Maturity Model)[5] claiming that greater maturity should mean greater business benefits. Organizations use software systems to collect and manage their data and there exist maturity and capability models

978-1-4799-4173-5/14 $31.00 © 2014 IEEE
DOI 10.1109/IIAI-AAI.2014.89

for software process such as CMMI and ISO/IEC 15504 (SPICE: Software Process Improvement and Capability Determination). In this section, we briefly explain the CMMI models from the view point of system development.

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, and the models are developed and maintained by the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU).

Currently, there exist three specific models in CMMI, 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 required components in CMMI are the specific and generic goals. The expected components are the specific and generic practices describing the activities that are important in achieving a required CMMI component. The expected components in CMMI are the specific and generic practices. The informative material plays an important role in understanding the model. The informative components in CMMI are subpractices, notes, references, goal titles, practice titles, sources, example work products, and generic practice elaborations.

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 precess area for requirements development by using continuous representation.

We briefly explain CMMI-DEV[6] in assuming organizations have necessary capabilities for institutional research by developing such products and services by themselves rather than by acquiring from a third party.

A. CMMI-DEV

The CMMI-DEV provides guidance for applying CMMI best practices in a development organization. The CMMI-DEV 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)

When focusing on the degree in the relationships of the related process areas, the Requirements Development (RD) process area has the highest degree. Figure 1 shows the connections to and from the RD process area, in terms of the model component, related process areas. This figure indicates the quality of requirements may have impact on the process areas in the Project Management and Support categories as well as process areas in the Engineering categories.

The purpose of RD is to elicit, analyze, and establish customer, product, and product component requirements.

Specific goals and practices 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

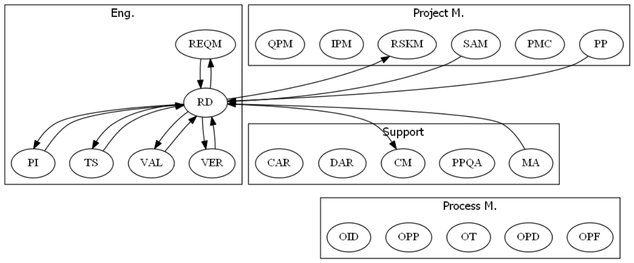

Figure 1. Related process areas of RD

- 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 acquires necessary capabilities for institutional research by developing such products and services, and tries to improve the RD process area in the CMMI-DEV.

III. Requirements Development

Developing requirements is important in acquiring systems effective in institutional research. 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.

A. Energy Management Needs and Wants 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[7]. The following are measures 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

The following are measures 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)
   - Keeping lid of the heating toilet seat closed and setting at the lowest temperature
4) Lifestyle
   - Abiding Japanese government campaign encouraging organizations to set their heater thermostats to the preset threshold over the winter.
   - 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 managers' point of view and an individual residents' point of view. We expect energy management systems cover these points to facilitate energy saving activities of all the members in our campus. However, our current energy management systems are basically building management systems and cover only building managers' 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 to develop proper requirements from needs and wants embedded in the descriptions explained above, in order to acquire proper energy management systems effective in promoting sustainability for whole stakeholders in our university.

Figure 2 shows overview of energy management systems in one of our campuses. Although we have energy management systems at a 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 3 shows a sample chart generated by one of our building energy management systems. Each vertical bar shows a total of energy consumption in a building in a day and it is difficult to make relationships between the activities of each member and their consequence from the graph.

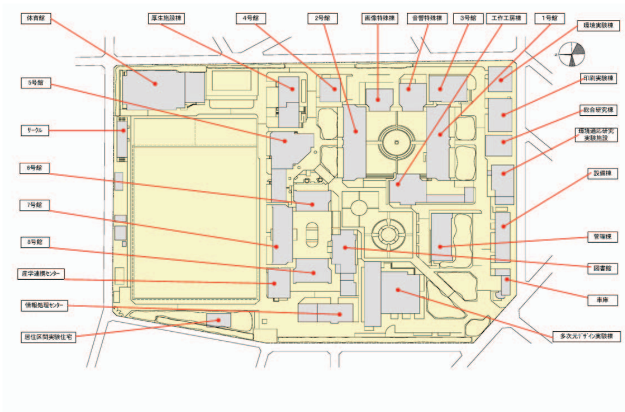

Figure 2. Overview of energy management systems of one of our campuses.

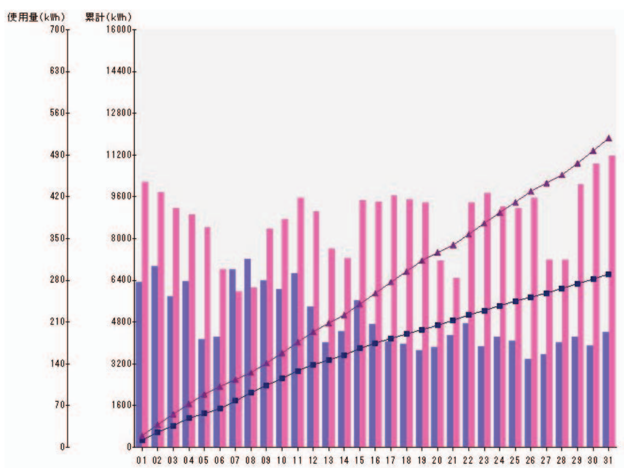

Figure 3. Sample chart generated by one of our building energy management systems.

We assume laboratory members are important stakeholders, and we put emphasis on a view point of laboratory as one of the primary administrative units in the university campus. Many of the students in certain grades and faculty members spend their time in their laboratories.

B. Software Product Line Engineering

Software product line engineering treat most software systems as not completely new ones but a family of software systems with commonalities and variabilities. This abstraction can be leveraged to improve the software system acquisition process. as well as the software system development process. In the software product line engineering paradigm[2], we have two separates processes as shown in Figure 4.

- Domain engineering process is responsible for establishing the reusable platform and for defining the commonality and the variability of the product line. The platform consists of various software artefacts ranging from requirements to tests.
- Application engineering process is responsible for deriving product line applications from the platform established in domain engineering process.

Energy management systems suitable for university campus seem different from other energy management systems such as home energy management system (HEMS) and building energy management system (BEMS), although they share a lot between them. We need to consider commonality and variability between energy management systems for smart campus and other energy management systems. As energy management systems are becoming software intensive like many other infrastructure systems, software product line engineering techniques seem to be becoming more important in developing energy management systems. We use domain engineering techniques in software product line in developing requirements of energy management systems for smart campus.

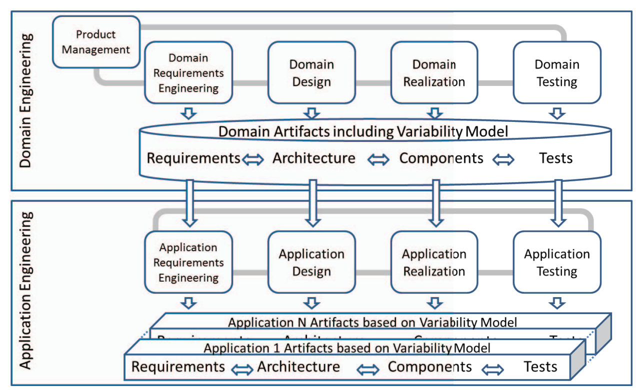

Figure 4. Outline of software product line engineering.

The following factors make requirements development of energy management systems for the smart campus difficult.

- There is a variation in development purposes and subjects among different energy management systems.
- Requirements, in different phases of the life-cycle, 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 exist some issues specific to the university campus.

- Each campus has unique characteristics as well as common characteristics.
- As the target unit of energy management, we have a variation in the logical point of view as well as in the physical point of view.

- 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 the university.

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

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

C. *Analysis for Commonalities and Variabilities in Requirements*

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. They 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 the laboratory energy management system in university campus.

From the view point of the 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 requirements to cover assumed scenarios by examining several HEMS and BEMS products as shown in Table I. We used our dictionary tool, as shown in Figure 5, in analyzing features of these energy management systems.

Table I
HEMS AND BEMS PRODUCTS WE ANALYZED.

	Product
HEMS A	Smart HEMS
HEMS B	NEC Cloud type
HEMS C	Dneso HEMS monitor
HEMS D	navi-ene
BEMS A	Panasonic ECO-SAS DR
BEMS B	enesys
BEMS C	enetune-bems
BEMS D	Azbil bems

We 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.

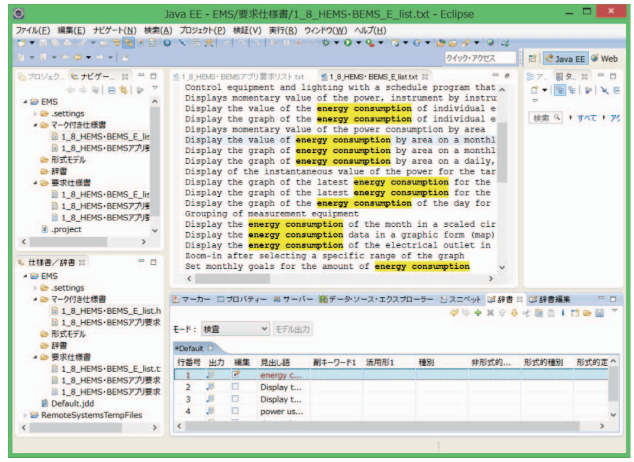

Figure 5. Snapshot of our dictionary tool in analyzing features of energy management systems.

The following are the core requirements.

- Display the energy consumption of the entire measurement area.
- Allow download of the measured data.
- Display the instantaneous power value.
- Set the upper limit of the instantaneous power value.
- Display the energy consumption of each equipment.
- Display the energy consumption of each user.
- Display the energy consumption of each category.
- Set equipment.
- Display a sorted list of the equipment.
- Allow user registration.
- Allow user configuration.

IV. Evaluation through Prototyping

As stated in the specific practice SP 3.5, Validate Requirements, in RD process area in CMMI-DEV, examples of techniques used for requirements validation include the following: Analysis, Simulations, Prototyping, and Demonstrations. We chose Prototyping and developed a prototype as explained below.

In order to deal with the variabilities as seen above, we use an incremental and manageable process. We strategically develop the target product through the multiple development cycles, starting 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.

As a prototype, we developed a system whose sample output is shown in Figure 6. In this system, we use energy consumption data and floor plans for laboratories. The output shown in Figure 6 focuses on energy consumption of a specific laboratory. This system seems more useful in increasing the awareness and involvement of laboratory members for smart energy management compared with more coarse grained BEMS-based output like Figure 3.

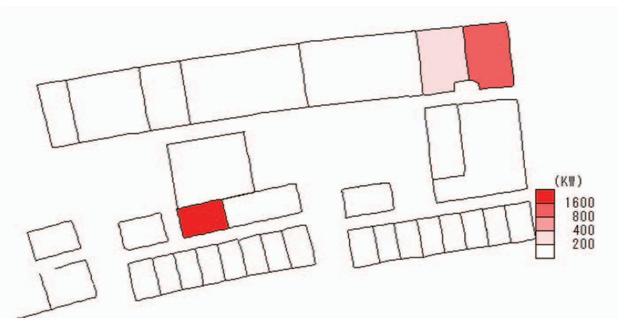

Figure 6. Sample output of a laboratory energy management system. This output focuses on energy consumption of a specific laboratory.

We have been studying the effectiveness of formal methods in software development process[8][9]. As an advance approach, we also examined the effectiveness of formal methods in requirement development. In this case study, we used a model-oriented formal method, VDM[10], in developing and analyzing requirements in an efficient but disciplined way. By using VDM, we could find defects on data types frequently used in energy management systems at a requirements level before going into the design. We confirmed positive impacts of the formal method from the view point of development process for the laboratory energy management system. We could generate code from a specification, but we need to specify some amount of glues to fill the gap between the formal specification language and the implementation language (Java, in this case). The cost of this additional task is the subject of our future work.

V. Concluding Remarks

We believe the capability and maturity of software engineering are also important for institutional research as it uses software-based systems to conduct analyses for decision making, planning and continuous improvement in educational institutes while collecting evidence, data, and information in educational institutes. In this paper, we discussed advanced software engineering technologies, such as software process improvement models and software product line engineering to develop systems effective in institutional research systems. We reported a trial case of requirements development in improving the energy management system in our campus. We could develop 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.

References

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

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

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

[4] DMBOK (Data Management Book of Knowledge) , http://www.dama.org/

[5] Enterprise Data Management Council Data Management Maturity Model, http://edmcouncil.org/

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

[7] Energy Saving Policy, Kyushu University Environmental Report 2013, Chapter 3 (In Japanese) http://www.kyushu-u.ac.jp/university/data/kankyohoukokusyo/2013/2013chapter-3.pdf

[8] S. Kusakabe, Y. Omori and K. Araki: A Combination of a Formal Method and PSP for Improving Software Process, TSP Symposium 2012, CMU / SEI-2012-SR-015, 2012.

[9] S. 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).

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