A Light-Weight CMMI Assessment Approach by Sequential Pattern Mining

Nien Lin Hsueh
Department of IECS
Feng Chia University
nlhsueh@fcu.edu.tw

Chia Chun Kuo
Department of IECS
Feng Chia University
yaboe788@ms8.hinet.net

Lien-Fu Lai
Department of CSIE
National Chunghua
University of
Education, Taiwan

Jong-Yih Kuo
Department of CSIE
National Taipei
University of
Technology

Abstract

CMMI is now the de-facto industrial standard for process improvement. SCAMPI is a benchmark quality rating method relative to CMMI model. As SCAMPI appraisal is a fully human-judged, instantiation-based method that will take the organization a lot of time, in this paper we propose a light-weight and mining-based approach called MBGA (mining-based goal analysis) to help organization evaluate the gap between the organization and the CMMI goals.

Keyword: CMMI, Data Mining, Soft Goal

1. Introduction

Capability Maturity Model Integration, CMMI [1], is the new de-facto standard process improvement model for determining the organizational maturity in product or software development. CMMI is an important model for product development industry. Many procurers require a specific level of maturity from their suppliers. Also many companies set internal process improvement objectives driven by the maturity levels of CMMI.

CMMI defines the maturity levels through process areas. By default, every organization is at maturity level 1. To reach level 2, an organization should satisfy the goals of seven process areas – such as *Requirements Management* and *Project Planning*. To achieve the level 3, an organization should perform all the process areas of the level 2 plus the process areas defined for the level 3 – such as *Requirements Development* and *Technical Solution*. Analogically, maturity levels 4 and 5 require the implementation of new process areas as well as those of the lower level process areas.

To provide a benchmark quality rating relative to CMMI, the Software Engineering Institute (SEI) at Carnegie Mellon University develops a standard appraisal method called SCAMPI (Standard CMMI Appraisal Method for Process Improvement) [2]. SCAMPI relies on an aggregation of evidence that is collected via instruments, presentation, documents, and interviews. The appraisal team observes, hears and reads these evidences, and then transforms them into notes, and then into statements of practices of practice implementation gaps or strengths, and then preliminary finds. The process is well organized into a three-phase process: plan and prepare for appraisal, conduct appraisal and report results.

Though SCAMPI is a very rigorous method, there are some weaknesses: (1) it takes a lot of effort. The actual effort depends on the number of projects selected, the size of the organization unit, and the involved stakeholders. Normally, it takes 10 days for a maturity level 3 appraisal; (2) the assessment is subjective to appraisal team members (or called ATM). The ATM maybe consisted of organization members, therefore the result is not objective; and (3) the goals are ranked either satisfied or not satisfied- which can not represent the real organization situation. An organization maybe ranked as “not satisfied” even they implement most practices.

In this paper, we assume the organization has developed a set of organization processes for CMMI and accordingly developed a workflow system for the processes. When members are using the workflow system for their projects, all logs will be recorded. We utilize the sequential data mining technique to analyze the log, and then evaluate its goal achievement. In addition, the goal in our model can be satisfied to a degree. The benefits of our approach are

- Organization members can spend less effort on process assessment, which reduce the resistance to perform continuous process improvement.
- The goal achievement is evaluated as a degree value, which can reflect the gap to CMMI maturity/capability more explicitly.

This paper is organized as follows. Section 2 describes the CMMI basic structure. Section 3 introduces SCAMPI approach. Section 4 introduces our main idea and approach- including the data mining technology and its application on our mining-based gap analysis. In section 5 is our conclusion.

2. CMMI Basic Structure

The core components of CMMI are process area, goal and practice. A process area is consisted of many process-area-specific specific goals and process-area-independent goals. A specific goal describes the unique characteristics that must be present to satisfy the process area, whereas the generic goal appears in many process areas. For example, *Establish Estimates* and *Develop a Project Plan* are two specific goals defined in *Project Plan (PP)* process area. *Institutionalize a Managed Process* is a generic goal defined in all process areas.

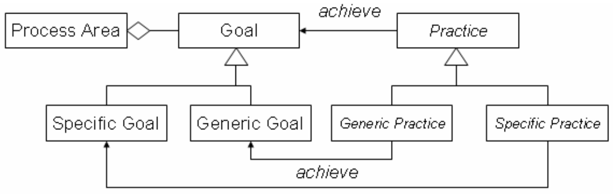


Figure 1. Process Area, Goals and Practices

Practice is the description of an activity that is concerned important in achieving the associated specific goal. *Specific Practice* describes the activities expected to result in achievement of specific goal, whereas the *generic practice* describes the activities to achieve the generic goals. For example, to achieve the goal *Establish Estimates*, CMMI defines a practice: *Estimate the Scope of the Project*.

The practice defined in a goal does not constrain the techniques you apply in your project or organization. For instance, you can apply *Line of Code* (LOC) or *function point* as the size measure when you estimate your work product. The practice is also methodology- independent – you can use SA/SD approach or OOA/OOD approach when realizing the goals in *Technical Solution (TS)*.

3. SCAMPI Approach

In the CMMI model, there is a direct relationship between goals and practices that contribute forward achievement of those goals. In the SCAMPI method, a fundamental premise is that satisfaction of goals can be determined only upon detailed investigation of the extent to which each corresponding practice is implemented, for each sample instance used as a basis for the appraisal. And the aggregate of objective evidence provided is used as the basis for determining practice implementation. Therefore, SCAMPI defines a rating process:

- **Instantiation-level practice characterization.** On this phase, we assign a value to describe the extent to which CMMI model is implemented for a specific instantiation (project). The range of value for practice characterization values includes Full Implemented (FI), Largely Implemented (LI), Partially Implemented (PI) and Not Implemented (NI). Table 1 summarizes rules for characterizing instantiation-level implementations of practices. Practice characterization values are assigned to each CMMI model practice for each process instantiation within the appraisal scope, and aggregated to the organizational unit level.
- **Organization-level practice characterization.** After characterizing the implementation of model practices, the practice implementation characterization values will aggregate to the organization unit level. The extent to which an organizational unit has implemented CMMI model practices can be determined only by considering, in aggregate, the extent to which those practices are implemented by instantiations of process. Table 2 summarizes rules for aggregating instantiation-level characterizations to derive organizational unit-level characterizations.
- **Goal satisfaction ratings.** In this phase, the aim is to create goal-level statements that summarize the gaps in practice implementation. These statements must be abstracted to the level of the organizational unit, and cannot focus on individual projects. And if there are no findings that document the weaknesses associated with a goal, the goal must be satisfied. That is, the goal is rated satisfied if
  - All associated practices are characterized at the organizational unit level as either LI or FI, and
  - The aggregation of weaknesses associated with the goal does not have a significant negative impact on goal achievement.

  If a goal is rated as unsatisfied, the team must be able to describe how the set of weaknesses led to this rating.

Table 1. Rules for characterizing practice implementation

Characterization	Meaning
Full Implemented (FI)	• The direct artifact is present and judged to be appropriate. • At least one indirect artifact and/or affirmation exist to confirm the implementation. • No substantial weaknesses were noted.
Largely Implemented (LI)	• The direct artifact is present and judged to be appropriate. • At least one indirect artifact and/or affirmation exist to confirm the implementation. • One or more weaknesses were noted.
Partially Implemented (PI)	• The direct artifact is absent or judged to be inadequate.

	• Artifacts or affirmations suggest that some aspects of the practice are implemented. • Weaknesses have been documented.
Not Implemented (NI)	• Any situation not covered above.

Table 2. Rules for organization-level characterizations

Condition	Outcome	Remarks
All X (e.g., all LI)	X	All instantiations have the same characterization.
All LI or FI	LI	All instantiations are characterized LI or higher.
Any PI, No NI	LI or PI	Team judgment is applied to choose LI or PI for the organizational unit.
Any NI	NI, PI, or LI	Team judgment is applied to choose NI, PI or LI for the organizational unit.

The introduction of above description describes the concept of SCAMPI. According to SCAMPI, organization can rate the maturity level.

4. Mining-Based Goal Analysis (MBGA)

4.1. Basic Idea

An organization defines a process *op* for implementing the practice *p*. Where *op* = <A_1, A_2, A_3 ... *Ai*>, a sequence of activities A_i. The organization defines a set of processes *OP* for implementing all practices required for a specific maturity level, and a workflow system *WF(OP)* to implement the processes in *OP*. Basically, for the process *op* = <A_1, A_2, A_3, ... A_i >, the system must provide the functionality <f_1, f_2, f_3, ... f_i > respectively.

When using the *WF(OP)*, performing the functionality f_i, some log will be generated. *Log*(f_1) denotes the log for performing the f_1 functionality. *Log(WF(OP))* denotes all the logs when using the processes for a time period. By analyzing *Log(WF(OP))*, we want to know whether the organization still achieve the goals defined in the maturity level.

To make sure the all *WF(OP)* users obey the process, we can add constraint into *WF(OP)* such that the user cannot perform f_i without performing f_{i-1}. In this case, the organization either satisfies all goals or it fails to perform the process since the process is rigorously embedded into the workflow system.

However, sometimes the constraint is too strong and makes the process impractical for an emergent mission. In this case, we "soften" the constraint and generate a little different process called *op-variant*. The *op-variant* may be like <A_1, A_3, A_4, A_5>. Note that the A_2 is omitted from the process. Therefore, for each practice *p*, now we may define a set of *op* for it. *p(op)* is the set of *op* for the practice *p*, which at least contains an *op* that completely implements *p*, and various *op-variant* that implement *p* to a certain extent. We call the degree that a *op* (or *op-variant*) implements a practice as "practice implementation degree", denoted as *PI(op,p)*. The degree is ranged from 0 to 1. If an organization performs too many low-degree *op*, the gap to the maturity will be large.

We use an example to explain the idea again. If *p(op)* = {op_1,op_2,op_3} where *PI*(op_1,*p*)=1, *PI*(op_2,*p*)=0.8, *PI*(op_3,*p*)=0.5. It means that op_1 is the process for fully implementing the practice *p*, whereas op_2 and op_3 are *op-variant*. The op_2 is much better than op_3 in the sense that it can realize *p* much better. However, op_3 is less-constricted and easy to perform. And then, we redesign (extend) our *OP* to *OP'* which includes the original *OP* and their op-variants. The new workflow system is denoted as *WF(OP')*. Now, by analyzing the *Log(WF(OP'))*, we intent to analyze the achievement of each goal. The general analysis process is:

- Determining *typical organization process*: Typical organization process is the most frequently used for each practice. In the previous example, we analyze *Log(WF(OP'))* by mining techniques to determine op_1, op_2 or op_3 is the typical organization process. It means that op_1, op_2 and op_3 are our designed processes to implement the practice *p*.
- Getting practice implementation degree. For each practice, we get its practice implementation degree in this step. The practice implementation degree is equal to *PI(typical organization process)*.
- Calculating adjusted practice implementation (*API*) degree. *API* is the practice implementation degree after concerning the dependencies between practices.
- Evaluating Gap. First, we calculate goal satisfaction degree by aggregating all its adjusted practice implementation degree, and the evaluate the gap based on the goal achievement degree.

In the next subsections, we step by step describe our approach in detail. Figure 3 is a reference example for our approach.

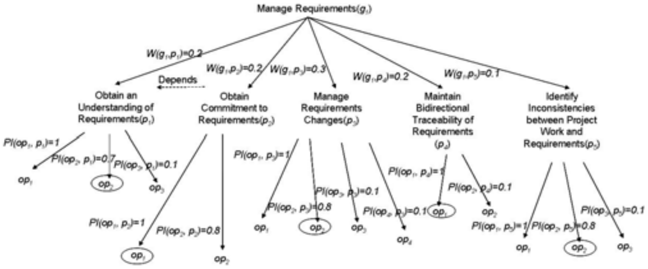


Figure 3. The relation between CMMI manage requirements goal and organization process

4.2. Determining Typical Organization Process

Data mining also referred to as knowledge discovery in databases, is a relatively new research area that aims at extracting previously unknown and potentially useful knowledge from large databases. One of the most important data mining problems is discovery of frequently occurring patterns in sequential data. Sequential pattern mining mainly used to analysis some data correlated with order, discover frequent sub-sequences in sequence databases [3, 4, 5, 6].

Because activities within the organization process have an order relation, in this work we apply data mining to organization process. We hope to find out the relevant stakeholders' behavior from the usage log. According to the behavior, we try to evaluate goal achievement defined CMMI.

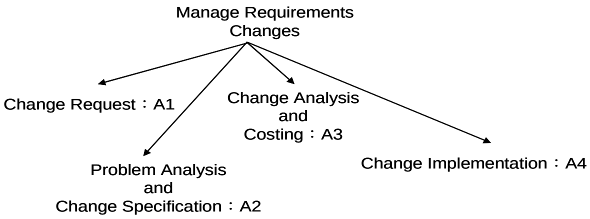

Manage Requirements Changes
Change Request：A1
Change Analysis and Costing：A3
Problem Analysis and Change Specification：A2
Change Implementation：A4

Figure 4. Activities of manage requirements changes

According to CMMI basic structure, software engineering discipline [7, 8] and organization culture, an organization can define different activity sequences for a process and their respective practice implementation degree to implement the practice, for example, *Manage Requirements Changes*. Let's assume we have a set of processes for a practice:

$1. op_1 : A1 \rightarrow A2 \rightarrow A3 \rightarrow A4$

$2. op_2 : A1 \rightarrow A3 \rightarrow A4$

$3. op_3 : A1 \rightarrow A2 \rightarrow A4$

$4. op_4 : A1 \rightarrow A4$

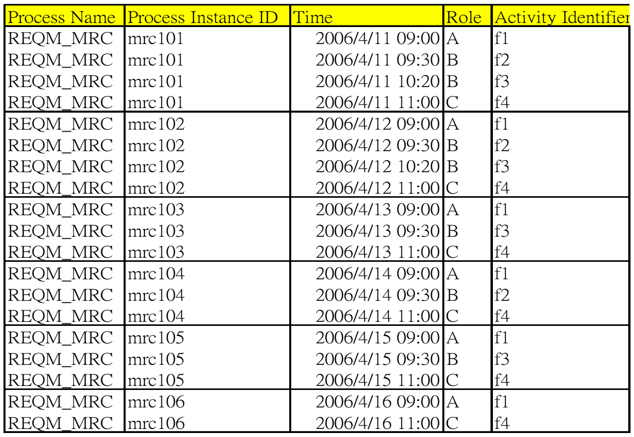
$$\begin{aligned} where\ PI(op_1, manage\ requirements\ changes) &= 1 \\ PI(op_2, manage\ requirements\ changes) &= 0.8 \\ PI(op_3, manage\ requirements\ changes) &= 0.1 \\ PI(op_4, manage\ requirements\ changes) &= 0.1 \end{aligned}$$

Note that the op_2, op_3 and op_4 are *op-variant* of op_1 since their PI values are not 1.

After organization processes are defined, the relevant stakeholders begin using workflow system to develop software products. When relevant stakeholders using workflow system to develop software products, there are some logs will be recorded. The format of the log is defined as below such that we can derive what is the typical process in the organization:

- *Process Name*: the CMMI practices correspond to the functionality f_i of *WF(OP')*.
- *Process Instance ID*: the organization process instance ID corresponds to the functionality f_i of *WF(OP')*.
- *Time*: the time that a stakeholder starts using an function f_i.
- *Role*: Record which role uses this function f_i.
- *Activity Identifier*: the activity identifier (ID), for example, f_1, f_2, f_3 ... f_i.

Table 3. Initial data log



Process Name	Process Instance ID	Time	Role	Activity Identifier
REQM_MRC	mrc101	2006/4/11 09:00	A	f1
REQM_MRC	mrc101	2006/4/11 09:30	B	f2
REQM_MRC	mrc101	2006/4/11 10:20	B	f3
REQM_MRC	mrc101	2006/4/11 11:00	C	f4
REQM_MRC	mrc102	2006/4/12 09:00	A	f1
REQM_MRC	mrc102	2006/4/12 09:30	B	f2
REQM_MRC	mrc102	2006/4/12 10:20	B	f3
REQM_MRC	mrc102	2006/4/12 11:00	C	f4
REQM_MRC	mrc103	2006/4/13 09:00	A	f1
REQM_MRC	mrc103	2006/4/13 09:30	B	f3
REQM_MRC	mrc103	2006/4/13 11:00	C	f4
REQM_MRC	mrc104	2006/4/14 09:00	A	f1
REQM_MRC	mrc104	2006/4/14 09:30	B	f2
REQM_MRC	mrc104	2006/4/14 11:00	C	f4
REQM_MRC	mrc105	2006/4/15 09:00	A	f1
REQM_MRC	mrc105	2006/4/15 09:30	B	f3
REQM_MRC	mrc105	2006/4/15 11:00	C	f4
REQM_MRC	mrc106	2006/4/16 09:00	A	f1
REQM_MRC	mrc106	2006/4/16 11:00	C	f4

Table 3 represents the logs when relevant stakeholders using workflow system to develop software products. After recording these logs, we can use data mining tool that support the sequential pattern mining to help us finding the relevant stakeholders' behavior. In this work we use the data mining tool that developed by SIM project—service mining system [9, 10]. Table 4 represents the mining result whose minimum support is 50%, the "Pattern" represents the functionality execution order relation, and the "Support" represents the occurrence number in database. The Pattern length is the longest (in Table 4, (f1) means that length is 1, (f1)(f2) means that length is 2 , and so on) and its Support value is the highest (in Table 4, the right column represent each pattern's support) means the most frequently used process for relative practice. The most frequently used process is the typical organization process.

In the example of requirement management, shown as Figure 3. Suppose that by using the data mining tool, organization finds that the processes op_2, op_1, op_2, op_1, and op_2 are the most frequently used process for the practices p_1, p_2, p_3, p_4 and p_5 respectively. Therefore, the processes op_2, op_1, op_2, op_1, and op_2 are the typical organization processes to implement the practices p_1, p_2, p_3, p_4 and p_5 respectively.

4.3. Getting Practice Implementation Degree

After the typical organization process is found out, it is very easy to derive the practice implementation degree. In our approach, a practice implementation degree for a CMMI practice p_i is the defined as the practice implementation degree to the typical organization process, that is, $PI(p_i, op_t)$ where op_t is the typical process that we found out in the mining phase.

As described in 4.2, the degree *PI(*op_j*, manage requirements changes)* for all *j* is predefined before the organization uses the workflow system. Therefore, *PI(*op_t*, manage requirements changes)* can be derived directly. In the previous example, as we have identified op_2 is the typical process, therefore we can know the practice implementation for *manage requirements changes* is 0.8 since *PI(*op_2*, manage requirements changes)* is 0.8.

In the example of requirement management, the practice implementation degree for p_1, p_2, p_3, p_4 are 0.7, 1, 0.8, 1 and 0.8 respectively.

Table 4. Data mining result

Pattern	Support
(f1)	6
(f1)(f2)	3
(f1)(f2)(f4)	3
(f1)(f3)	4
(f1)(f3)(f4)	4
(f1)(f4)	6
(f2)	3
(f2)(f4)	3
(f3)	4
(f3)(f4)	4
(f4)	6

4.4. Calculating Adjusted Practice Implementation Degree

Definition: *Dependency between practices. Assume* p_i, $p_j \in P(g)$, *P(g) represent the set of practices defined in CMMI to achieve the goal g,* p_i *may depend on* p_j *in the sense that* p_i *can't be implemented without the fully implementation of* p_j*. The dependency is denoted as depends(*p_i*,*p_j*).*

Referring to the process area *Requirement Management*, there is a goal in this process area — *Manage Requirements*. This goal includes five practices to achieve (as show in figure 4). Though CMMI does not define the dependency, we can derive the dependency between these practices after understanding their semantics. For example, there is a dependency between the *Obtain an Understanding of Requirements (OUR)* and *Obtain Commitment to Requirements (OCR)*. *OUR* deals with reaching an understanding with the requirements providers; *OCR* deals with agreements and commitments among those who have to carry out the activities necessary to implement the requirements. It is very intuitive to induce *OCR depends OUR* since it does not make sense (i.e., it is not a reasonable requirement management policy) if we obtain commitment without understanding the requirements.

Therefore, even we have derived a PI value for a practice p_i in the previous step, its PI value may be meaningless if p_i depends on another practice p_j and $PI(p_j)$ is not fully implemented. We thus define the *Adjusted Practice Implementation Degree*, denoted as $API(op_i,p_j)$.

$$API(op_i, p_j) = \begin{cases} PI(op_i, p_j)\ ,\ if\ \forall dependency\ \ depends(p_j, p_i)\ ,\ API(op_t, p_i) = 1\ , \\ \qquad where\ op_t\ is\ the\ typical\ organizati\ on\ process\ to \\ \qquad implement\ \ p_i\ or\ \neg\exists\ practice\ \ p_i, depends(p_j, p_i). \\ 0\ ,\ if\ \exists practice\ \ p_i\ ,\ depends(p_j, p_i)\ ,\ API(op_t, p_i) \neq 1\ , \\ \quad where\ \ op_t\ is\ the\ typical\ \ organizati\ on\ \ process\ \ to \\ \quad implement\ \ \ p_i. \end{cases}$$

In the example of requirement management, because the practice p_1, p_3, p_4 and p_5 do not depend on other practices, the adjusted practice implementation degree is equal to practice implementation degree. In the case of p_2, since p_2 depends on practice p_1 and the adjusted practice implementation degree of p_1 is not equal 1, so the adjusted practice implementation degree of p_2 is 0.

4.5. Evaluating Gap

Even within the same goal, different practices have different important degrees to the goal. It means, for example, implementing p_i makes the goal more satisfied than implementing another practice p_j. We thus define the Practice Weight for a *<goal, practice>* pair.

Definition: *When a practice* $p_j \in P(g)$ *is fully implemented, the goal g can be satisfied with a degree in [0,1].The degree is called Practice Weight, denoted as W(g,*p_j*).*

Note that the summation of all practice weight under a goal must be 1, that is,

$$\sum_{j=1}^{n} W(g, p_j) = 1,\ where\ p_j \in P(g)$$

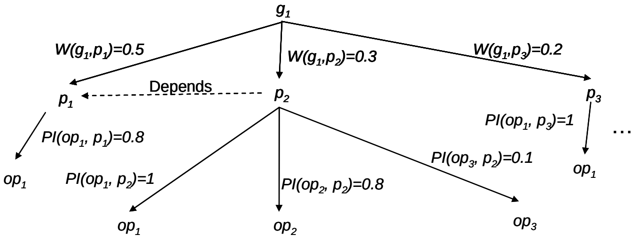


Figure 5. Practice weight with respective to a goal

We use the notation *Gap(g,OP')* to represent the gap between a CMMI goal *g* and our defined set of organization processes *OP'*. In SCAMPI, the goals are ranked either *satisfied* or *not satisfied*. We use numerical method to represent the gap. If the gap is l, it means the goal is completely not satisfied; whereas if the gap is 0, it means the goal is completely satisfied.

Definition: *Gap between goal and organization process.*

$$
\begin{aligned}
Gap(g, OP') &= Gap(g, Log(WF(OP'))) \\
&= 1 - \sum_{i=1}^{n}\left[W(g, p_i) \times API(op_t, p_i)\right], \\
&\text{where } op_t \text{ is the typical organizati on process of } p_i \\
&\text{from mining } Log(WF(OP')).
\end{aligned}
$$

Note that the gap is derived on the basis of Practice Weight and Practice Implementation Degree for each practice. The more important of a practice is, the more critical it is for satisfying a goal and reducing the gap. If an important practice (high practice weight) is not implemented very well (low PI value), the gap to the CMMI goal will be large. Also note that the gap is based on mining the usage log since op_t is identified from mining the *Log(WF(OP'))*.

In the example of requirement management, after determining the practice weight, then the organization can evaluate their gap to CMMI, the evaluated result as follow:

$$
\begin{aligned}
Gap(g,OP') &= 1 - [(0.2)\times(0.7)+(0.2)\times 0+(0.3)\times 0.8+(0.2)\times 1+(0.1)\times 0.8] \\
&= 0.34
\end{aligned}
$$

In our research we have implemented a gap analyzer called Mining-based CMMI Gap Analysis Tool. By using this tool, we can help organization to analyze the gap automatically.

5. Conclusion

Continuous process improvement is the core of software process improvement. After an organization gets his maturity/capability certification, it is very important to keep the improvement mechanism – they must conduct the appraisal periodically, propose improvement plan and review the results. However, conventional appraisal takes a lot of effort, and the result is subjective to human. To resolve the problem, we propose a light-weight and mining-based approach called MBGA to help organization keep the improvement mechanism. Because MBGA will collect analyzed data and execute gap analysis automatically, by using our approach, it can help the organization reduces effort and time cost in the appraisal.

Acknowledgements

The research was supported by Ministry of Economic Affairs (Taiwan) under grants 95-EC-17-A-02-S1-029.

6: References

[1] M.B. Chrissis, M. Konrad and S. Shrum, *CMMI: Guidelines for Process Integration and Product Improvement*, Addison-Wesley Professional, 2003.

[2] D.M. Ahern, J. Armstrong, A. Clouse, J.R. Ferguson, W. Hayes and K.E. Nidiffer, *CMMI SCAMPI Distilled: Appraisals for Process Improvement*, Addison-Wesley Professional, 2005.

[3] M.S. Chen, J. Han and P.S. Yu, "Data mining: an overview from a database perspective", *IEEE Transactions on Knowledge and Data Engineering*, Dec. 1996, pp. 866-883.

[4] R. Agrawal and R. Srikant, "Mining sequential patterns", *Data Engineering, 1995. Proceedings of the Eleventh International Conference*, Taipei, 1995, pp. 3-14.

[5] Y.L. Chen, M.C. Chiang and M.T. Ko, "Discovering time-interval sequential patterns in sequence databases", *Expert Systems with Applications*, October, 2003, pp. 343-354.

[6] B. Zhou, S.C. Hui and K. Chang, "An intelligent recommender system using sequential Web access patterns", *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, Dec. 2004, pp. 393 - 398.

[7] The Fundamental Rules" of Software Engineering URL: http://www.ics.uci.edu/~emilyo/SimSE/se_rules.html

[8] I. Sommerville, *Software Engineering*, Addison-Wesley, 2001.

[9] Service-oriented Information Marketplace, SIM. URL: http://140.115.51.135/index.html

[10] K.Y. Huang and C.H. Chang, "SMCA: A General Model for mining Asynchronous Periodic Pattern in temporal database", *IEEE Transaction on Knowledge and Date Engineering*, June 2005, pp. 774-785.