Modeling Framework and Supporting System for Process Assessment Documents

Makoto Matsushita[1], Hajimu Iida[2], and Katsuro Inoue[1]
[1]Osaka University, Osaka, JAPAN
[2]Nara Institute of Science and Technology, Nara, JAPAN

Abstract

Process is usually assessed by comparing with the assessment documents, though the information for the assessments are scattered; it is a time-consuming job to gather them. In this paper, we investigated the method to obtain information for process assessment by means of SPICE (Software Process Improvement Capability dEtermination). First, we extracted the information related to processes, products, and levels. Then, to construct the SPICE documents based on that extracted information, SGML (Standard Generalized Markup Language) was adopted. Based on the constructed SPICE documents, we have made prototypes of two tools, a tool that shows the information about process assessment on the display, and a tool that investigates relations between processes and products. These tools enable us easily to get information for process assessment, and to improve process assessments greatly.

1. Introduction

Improving software development processes are the important issues to achieve effective software production or to reduce the cost of software development. To improve this, first we should evaluate what the target of software development process is going on.

Recently there are various studies of software process assessment and software quality assurance, and the fruits are widely used in software development organization [4,8]. There are lots of evaluation methods and reference model, including CMM (Capability Maturity Model) [6,7], of SEI, ISO-9000 series

standards [14], SPICE (Software Process Improvement Capability dEtermination) [17], etc.

Software process is usually assessed with examining the documents of the projects or having interviews with the engineers and managers, by the experts of software process assessment standards. However, such procedure is a time-consuming job to execute, and the costs of this are very large; it seems that it is difficult to do [4,11].

In this paper, we have designed the simple model that does not introduce our original interpretation of software process assessment standards. The model consists of three elements and relationships between them. The model is described with SGML (Standard Generalized Markup Language) tag. We reorganized software process assessment documents as SGML documents. Using these documents, these standards can be easily handled formally and implemented easily. We have also designed a process assessment supporting system for self-assessment, and implemented two process assessment supporting tools.

2. SPICE

SPICE is one of software process assessment standards[1], and now it is standardized by ISO. The whole document of SPICE is about 400 or more pages. SPICE arranges the whole activities in software development environment into five “process categories”. Each process category consists of a set of “processes”, and process is consists of a set of “base practice”. SPICE has yet another axis of activities for evaluating the capability for each software development activity named “capability level”. Each capability level consists of a set of “common features” which represent the characteristics of activities. Each “common features” consists of a set of “generic practice” (fig.1) [21].

[1] Current version of SPICE was changed described in this paper. However, our proposed framework is independant from old SPICE specification; we think that adapting current SPICE to our framework should be possible.

The rough procedure to evaluate a software development process has three steps; first, it should be decided what to be evaluated. Then, information is gathered from the target process. Finally, information is evaluated checking with the standards to sum up the result [20].

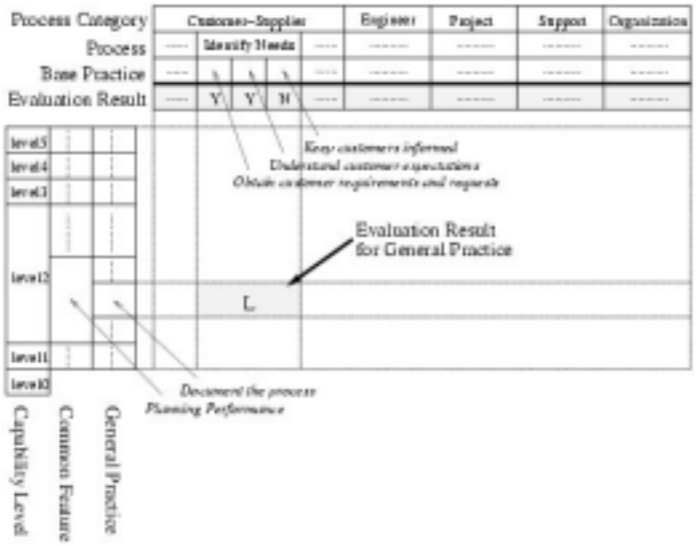

Figure 1: The classification of activity and the decision of assessment in SPICE framework

3. Generic Modeling for Software Process Assessment Standards

In this section, we show our modeling approach of generic software development process in the assessment standards with the SPICE framework.

3.1 Modeling Policy

The whole of software development activities focused in the software process assessment standards are modeled with following three types of elements and four types of relationships of these elements.

Elements

- Task: Task represents a set of activity in software development environments. Process categories, processes, base practices in SPICE will be candidates.

- Level: Level represents the achievements of software development work. Capability levels, common features, and generic practices in SPICE will be candidates.

- Product: Product represents generated materials and/or documents when tasks go on, or prepared when the tasks start.

Relationships

- Task – Product: Relationship about “a product is needed to enact a task”.

- Task – Task: Relationship about “an another task is performed to enact a task”.

- Level – Task: Relationship about “a task is performed to achieve a level”.

- Level – Product: Relationship about “a product is needed to achieve a level”.

The elements defined in our model can be extracted from the elements of SPICE standards, and the relationships can be extracted from the descriptions of SPICE standards. Actual description of our model is shown in section 3.2.

3.2 Model Description with SGML

In our approach, the model proposed in section 3.1 is described with SGML (Standard Generalized Markup Language) tags that are inserted into the original documents. In general, formed documents are structured as SGML documents; it enables to process documents, to build documents databases, to exchange document formats [10].

We define two tags to markup the documents, ELEMENT and RELATION for elements and relationships of our model, respectively. The information of each element and relationship is described as attributes of these tags, and it represents clearly the meanings written in software process assessment standards. We also define other tags to represent the structure of document itself, including the preamble, a name of base practice, etc. Figure 2 shows a fragment of document.

```
<ELEMENT TYPE=TASK ID="CUS.1" SUBID="2">
<PREAMBLE>
CUS.1.2
</PREAMBLE>
<TITLE>
Define the requirements.
</TITLE>
<BODY>
Prepare the system and software requirements
to satisfy the need for a new product and/or
service.  Note: This definition of the requirements
may be done completely or partially by the supplier.
<RELATION TYPE=TKTK SRC="CUS.1.2"
                DST="ENG.1" DST="ENG.2">
See "Develop System Requirements and Design"
ENG.1 and "Develop Software Requirements" ENG.2
</RELATION>
<RELATION TYPE=TKTK SRC="CUS.1.2"
                DST="CUS.3.1">
Also see CUS.3.1, "Obtain customer requirements and
requests." CUS 1.2 is focusing on defining requirements
when the software organization is acting as a customer.
CUS.3.1 is focusing on obtaining requirements when the
software organization is acting as a supplier. The primary
difference is one of perspective, the role being performed.
</RELATION>
</BODY>
</ELEMENT>
```

Figure 2: An example of a SPICE document with SGML

4. Supporting System

This section describes an experimental supporting system for software process assessment. The system employs SPICE as an assessment standard, and uses tagged documents, which describes in the previous section. The purpose of this system is to evaluate own software development process by developers or managers themselves.

The system supports to view the documents, find a relationship of documents, to apply logical operation, to maintain correspondences between the standard and actual software process, to show assessment information, and to register/accumulate assessment results.

The system consists of two tools and associated database (figure 3). There are two tools; a tool that investigates the information about process assessment document itself, elements and relationships written in the documents, and a tool which manages the result of software process assessment. Each tool uses SPICE documents that is tagged based on our model, and saves the analyzed result to the database. The result of software process assessment is also saved to the another database.

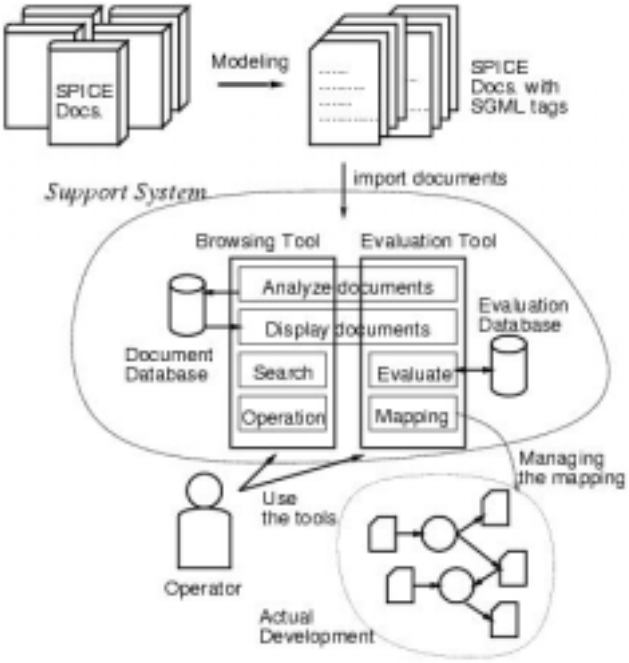

Figure 3: System overview

4.1 Document Viewer

The document viewer shows the information described in SPICE documents, the graphs of reference between tasks etc. Figure 4 shows a screen-shot of the document viewer. This tool has following features:

- Word searching: The tool shows a fragment of documents corresponding to a user's input.

- Keyword searching: The tool enumerates the name of task and/or products corresponding to a user's input. In addition, corresponding documents are shown by selecting enumerated one.

- Relation searching: The tool traverses the relationships in the documents. Traversing may be recursive, so derivative relations can be found.

- Logical operation: Operations described above can be combined each other by logical operations, such as getting intersection of two search results. Operation results can be *piped* to other operation's input.

Figure 4: The document viewer

The document viewer shows the result of these features visually. If the result will be a set, the tool shows the result as a list; if the result will be a tree structure, the tool shows the graph of the tree. In figure 4, the result of tasks that are referred from a task is shown as tree structure.

4.2 Evaluation Supporting Tool

The evaluation-supporting tool is for self-assessments of software process. Figure 5 shows a screen-shot of this tool. This tool has following features:

- Document viewer: It is a subset of document viewer tool previously shown. This tool shows a definition of words, task or products. If tasks are already evaluated, its results are also shown.

- Database for mapping the standards and an actual environment: The tool manages the relation between elements described in assessment documents and actual software development environment. These relations are sorted and showed in a table, and stored into a database. The files registered to database can be displayed.

- Database of assessment result: Users may enter the result of the assessment; specify a task, or a task and associated level, then enter the evaluation (two-level or four-level evaluation) with a dialog. Evaluation result can be selected with a button. These results are stored into a database.

- Collecting assessment results: The tool collects the result of evaluation, then sums up each process category with a table. In addition, the tool sums up each capability level and process category, then shows the results with their capability maturity percentages.

Figure 5: The evaluation supporting tool

We have implemented and evaluated a prototype of this system with a software development environment based on ISPW-6 [2].

5. Discussion

5.1 Modeling Approach

Our modeling approach described in section 3.1 can retrieve the information described in a large software process assessment documents such as SPICE. In general, designing an assessment supporting system requires own interpretation or explanation that is pre-defined by tool designer, then implements a system with this requirement [3]. This approach may include a wrong interpretation of assessment document to the system design, and may lead to a wrong result of assessment. Our approach uses the original definition or relationships written in the original document; it is less different from the original interpretation of assessment.

5.2 Tagged Document

In this work, we employ SGML as a description language of our model, design and implement a tool based on tagged documents. SGML is commonly used for various objectives such as reusing document or full-text database [5,9]. However, these applications are intended to handle many files in the same format. Our system handles large single file and provides a feature to operate a file. In addition, there are many SGML-based documenting support environments [12,15,16], and they become popular tools for electric documents. However, these environments have so many features and we want to keep the whole system to be simple.

Recently, WWW (World-Wide Web) and HTML (HyperText Markup Language [1]) are widely used in many places, especially in the Internet. There are many tools or environments for HTML and we may use these powerful tools. However, our purpose requires our own tag definition and implementation and it requires some extension to HTML format; it should lose the portability of HTML, so it should be avoided. We are now investigating an XML as our model representation format.

5.3 Supporting System

Our system is for the self-assessment, to support to give a hint of the process improvement. Such assessment methods are not verified and guaranteed by external organization, however, it is easy to execute with appropriate assessment criterion; software assessment activities are widely used in lots of software development organization.

It is a long term to proceed a software process assessment to an actual software development project. Our tools has evaluation results database, so that suspending and resuming the evaluation procedure is quite easy; it can be used for long term evaluation. Our tools also support a database for relation of assessment standards and actual environment, so it may supports the decision of evaluation.

Our current prototype supports two types of activities in software process assessment, gathering an information for evaluation, and calculation of the evaluation result. These features are designed and implemented with the procedures defined in an assessment standards [20], so users can do the right way to assess a target process. Software process improvement based on the assessment result will bring about better and effective software development. This prototype currently depends on SPICE standards. However, our model does not depend on it; adapting other standards to the prototype is possible.

6. Conclusion

We have proposed a model of software process assessment documents, and defined it as SGML documents. We also designed and implemented a prototype of software assessment supporting system. Using our approach and system, software assessment documents can be formalized easily and the result of system brings simple software assessment method.

As a further work, validation of our model and environment through experiments and applying other software process assessment standards to our model are planned. In addition, supporting other phases in process evaluation (preparation, improvement planning, etc.) is needed to our system.

References

[1] Berners-Lee, T. and Connolly, D.W., “Hypertext Markup Language - 2.0”, RFC1866, Massachusetts Institute of Technology Laboratory for Computer Science / The World Wide Web Consortium, ftp://ds.internic.net/rfc/rfc1866.txt, 1995.

[2] Kellner, M.I., Feiler, P.H., Finkelstein, A, Katayama, T., Osterweil, L.J., Penado, M.H. and Rombach, H.D., “Software Process Modeling Example Problem”, In Proceedings of the 6th Int. Software Process Workshop, pp.19-29, 1990.

[3] Omoto, N., Komiyama, T. and Fujino, K., “Software Process Assessment Support System SPATS”, IPSJ Technical Journal, 95-SE-102-28, pp.159--164, 1995.

[4] MacLennan, F. and Ostrolenk, G., “The SPICE Trials: Validating the Framework”, In Proceedings of the 2nd International SPICE Symposium, pp.109--118, 1995.

[5] Morita, U., Suzuki, M., Miyagawa, K. and Hamanaka, H., “A Trial For Development of DTD for "JOHO KANRI" and "JOHO KANRI" Full Text Database by Using HTML”, IPSJ Technical Report, 95-FI-37-2, pp.7--14, 1995.

[6] M. Paulk, B. Curtis, M. Chrissis, and C. Wever, “Capability Maturity Model for Software, Version 1.1”, Software Engineering Institute, CMU/SEI-93-TR-24, 1993.

[7] M. Paulk, B. Curtis, M. Chrissis, and C. Wever, “Key Practices of the Capability Maturity Model, Version 1.1”, Software Engineering Institute, CMU/SEI-93-TR-25, 1993.

[8] H. Saiedian, and R. Kuzara, “SEI Capability Maturity Model's Impact on Contractors”, IEEE Computer, Vol.28, No.1, pp.16--26, 1995.

[9] Takayanagi, Y., Sakata, H. and Tanaka, Y., “Full-text Database System Based on SGML”, IPSJ Technical Report, 93-CH-18-5, pp.35--42, 1993.

[10] Tanaka, Y., “Standardization of SGML”, Journal of IPSJ, Vol.32, No.10, pp.1118--1125, 1991.

[11] I. Woodman, and R. Hunter, “Analysis of Assessment Data from Phase 1 of the SPICE trials”, Software Process Newsletter, No.6, pp.1--6, 1996.

[12] DocIntegra, http://www.hitachi.co.jp/Prod/comp/soft1/open/docint.htm, Hitachi Ltd., 1995.

[13] ISO 8879, “Information Processing - Text and Office System - Standard Generalized Markup Language (SGML)”, 1986.

[14] ISO 9000-3 Guidelines for the Application of ISO 9001 to the Development, Supply, and Maintenance of Software, 1991.

[15] OLIAS, http://www.fujitsu.co.jp/hypertext/granpower/topics/olias/olias.html, Fujitsu Limited, 1996.

[16] Panorama Pro, http://www.sq.com/products/panorama/panor-fe.htm, SoftQuad Inc., 1996.

[17] The SPICE Project, “Software Process Assessment -- Part 1: Concepts and Introductory Guide”, Version 0.02, 1994.

[18] The SPICE Project, “Software Process Assessment -- Part 2: A Model for Process Management”, Version 0.01, 1994.

[19] The SPICE Project, “Software Process Assessment -- Part 3: Rating Process”, Version 0.01, 1994.

[20] The SPICE Project, “Software Process Assessment -- Part 4: Guide to conducting assessments”, Version 0.02, 1994.

[21] The SPICE Project, “Software Process Assessment -- Part 5: Construction, Selection and Use of Assessment Instruments and Tools”, Version 0.02, 1994.