A Typed Generic Process Model for Product Focused Process Improvement

Manoranjan Satpathy       Rachel Harrison

School of Computer Science, Cybernetics and Electronic Engineering
University of Reading, Reading RG6 6AY, UK
(M.Satpathy, Rachel.Harrison}@reading.ac.uk

Abstract. The motivation behind the idea of product focused process improvement is to make a process improvement program address certain product quality features in an explicit manner. The PROFES methodology [PROFES (2001)] describes such an improvement program through the notion of a PPD (Product-Process Dependency) repository. The typed generic process model (TGPM) [Satpathy (2000)] is a parametric template which relates each process attribute with the way it is likely to affect the quality of the intermediate as well as the end products. TGPM could be instantiated to generate process models for individual processes. This paper shows how the TGPM and the PROFES methodology both address product quality in a similar manner, and how they could be complementary to each other as regards to product focused process improvement.

Keywords: Process Models; Product focused process improvement; Software Quality; Software Metrics.

1 Introduction

Continuous process improvement is integral to all the major process models such as ISO 9001[ISO (1994)], CMM [Humphrey (1992)] etc. Clearly processes need to be improved, mainly because: (i) market requirements become more and more demanding with time (ii) technology changes, and further (iii) better software practices evolve with time. In this paper, by process we mean any software activity associated with the development and maintenance of software, from requirement analysis through to maintenance. Each such process can be a candidate for improvement. The most important objective for process improvement is the improvement of end-product quality. This is particularly important for products such as embedded systems. It is not
only that the number of candidates for improvement is large, but also each process has many aspects (or factors) that may need improvement. However, the cost of implementing improvement actions can be high. Further, an end-product has many quality factors that may be improved. So, the optimal choice is to choose a few of the end-product quality factors, find out which improvement actions can lead to the improvement of such product quality factors, and implement those specific actions. This is the philosophy behind product focused process improvement (PFPI) [PROFES (2001)].

Process models like the Capability Maturity Model (CMM), ISO/IEC 12207, ISO 9001/9000-3 etc. are not specific enough to cater to the needs of individual processes. There are certain process aspects (including the duality of process attributes which we discuss later) which are not addressed by any of the existing models. In order to alleviate these problems, we have defined a typed generic process model (TGPM) [Satpathy (2001)]. The generic model could be instantiated to be the customized model of any individual process, and this instantiation mechanism clearly spells out the idiosyncrasies of the individual process that need special attention. Furthermore, it makes the relationship between process model and product quality explicit. TGPM in conjunction with the Goal-Question-Metric (GQM) method [Basili (1994)] offers a good process assessment and improvement framework.

The aim of this paper is to show that the ideas behind TGPM and the PROFES methodology are very similar, and how both can be complementary to each other in achieving PFPI. The organization of the paper is as follows. Section 2 discusses the related work. Section 3 and 4 respectively introduces the TGPM and the PROFES methodology. In Section 5, we present a comparative study of both. Section 6 describes a case study and the Section 7 concludes the paper.

2 Related Work

ISO/IEC 9126 [ISO (1991)] describes a generic model for specifying and evaluating the quality of software products. Focusing on product quality alone may not guarantee that an organization will deliver products of good quality. So, based on an orthogonal view that improving the quality of a process will deliver products of good quality, many models have been developed. Prominent among them are the CMM and ISO 90001.

The GQM method [Basili (1994)] proposes a measurement plan for assessing the quality of entities like products, processes or people. It starts with a set of business goals and the goals are progressively refined through
questions to obtain metrics for measurement. The measured values are then interpreted in order to answer the goals. Application of Metrics in Industry (ami) [Pulford 1996] combines CMM and the GQM method, and the result is that it provides a complete framework for process improvement.

Focusing on either process quality or product quality alone is not sufficient. It has been suggested that a combined study of products and processes is necessary for process assessment and improvement [Bazzana (1993)]. Most of the process models fail to make the relationship between process models and product quality clear. TGPM makes such a relationship clear by taking the dual nature of process attributes into account. A brief description of TGPM will be presented in the next section.

The PROFES improvement methodology [PROFES (2001), Jarvinen (2000)] follows an orthogonal approach to process improvement. It first uses ISO 9126 to identify the subfactors in relation to product quality which need to be improved. It identifies appropriate improvement actions. Then an action plan is made and the plan is executed. Section 3 will discuss the PROFES methodology in more detail.

3 The typed Generic Process Model (TGPM)

Typed Products and Typed Processes:

A process is any software activity which takes a set of products as input and produces a set of products as output. Formally, it could be defined as a relation from a set of products to another set of products; the set of relations from m input products to n output products could be denoted by:

\[ IP_1 \times IP_2 \times \ldots \times IP_m \rightarrow OP_1 \times OP_2 \times \ldots \times OP_n \]

where \( IP_i \) and \( OP_j \) are the types of the i-th input product and the j-th output product respectively; \( \rightarrow \) denotes the relation operator and \( \times \) denotes the cartesian product operator. Table 2 gives examples of process types. For instance, the formal specification process takes a requirement document as input and produces a formal specification as output. More about the types of products and processes can be found in [Satpathy (2000)].

| rd: \( RD \) | // Set of requirement documents |
| spc: \( SPEC \) | // Set of Specifications |
| fs: \( FS \) | // Set of formal specifications |
| des: \( DES \) | // Set of designs |

Table 1. Types of some products
Internal Process Model:
Under the TGPM framework, each process has a formal internal structure [Satpathy (2000)]. A process may be composed of subprocesses. A subprocess is also a process in the sense that it takes a product set as input and gives out a product set as output. Whether a process should be decomposed into subprocesses or not is decided by the process designer. When considered from a process point of view, the inputs and outputs of subprocesses are called intermediate product sets. A subprocess which is not decomposed is called an atomic process which is in turn defined as a set of steps, like the steps of an algorithm. For example, Figure 1 shows the internal details of the testing process. Unit Testing is a subprocess of Testing Process, and White Box Testing and Black Box Testing are atomic processes. An organization is expected to have its own definition of the internal structure of a process.

![Figure 1. Internal structure of Testing Process.](image)

Definition of TGPM:
The TGPM is next defined in terms 8 factors; each factor is in turn defined by a set of subfactors. The definitions of some subfactors of TGPM are shown in Appendix A. For a detailed discussion on them, refer to [Satpathy (2000)]. Note that the definitions in the Appendix are generic in nature and so they should be used after instantiation.

The most important aspect of TGPM is that it defines attributes from a dual perspective. Consider, for instance, the understandability attribute. The two perspectives are (i) the concepts of the process should itself be understandable to the process executer and further (ii) the process should make its output
product set understandable. When we consider the formal specification process, the method of creating the formal specification and the formal specification language itself must be understandable to the specifier, and further the specification process must make the formal specification understandable to its users (through the use of comments, diagrams etc.). TGPM could be seen as a template with three parameters:

$$\text{TGPM}(\text{inp-prod-set-type}, \text{out-prod-set-type}, \text{application-domain})$$

where \text{inp-prod-set-type} is the types of the set of products that a process takes as input, \text{out-prod-set-type} is the set of products that the process produces as output, and application domain denotes safety-critical, real-time, business etc.

**Customization of the TGPM:**

The customization of TGPM proceeds in two steps: (i) the substitution step and (ii) the refinement step. In the substitution step, we substitute the parameters with their actual bindings. Let us take the example of the formal specification (FS) process whose type is: $RD \leftrightarrow FS$. The substitution is illustrated by the expression:

$$\text{TGPM} \left\{ \{RD\} / \text{inp-prod-set-type} \right\} \left\{ \{FS\} / \text{out-prod-set-type} \right\}$$

What the above expression signifies is that, all occurrences of the input product set in the definitions of the factors and the subfactors of the TGPM are substituted by RD (requirement document). Similarly, all occurrences of output product set are substituted by FS. Thus, at the end of the substitution step, we have a crude definition of each of the subfactors of the process concerned.

For the application-domain parameter, suppose we have a safety critical application. Now, with the knowledge that we are dealing with a FS process in a safety critical application, the refinement step refines the crude definitions that we have obtained after the substitution step. The result then will be the customized quality model for the FS process. The definitions of the subfactors in Appendix A have been instantiated for the FS process in Appendix B.

Let us consider the instantiation of the subfactor correctness for the FS process. Then the input product set type is \{RD\}, and the output product set type is \{FS\}. Further, let the application domain be a safety critical application. So, after the substitution step, we get:

**Correctness:**

(i) The degree to which the FS process ensures that the functionalities of the RD are transformed accurately to those in FS.
(ii) Is there any gap between what the FS process does and what it is
supposed to do? (Are the process steps of the process accurately
followed?).

In the refinement step, we know that a Specifier is the process executor for
the FS process. At the RD level functionality is understood to be a ‘feature’.
So, the above definitions are refined as:

(i) The degree to which the FS process ensures that the features of
the RD are transformed accurately to the functionalities in the FS.

(ii) Is there any gap between what the FS process does and what it is
supposed to do?

**Process Assessment and Process Improvement using TGPM**

The TGPM in conjunction with the GQM method can provide a complete
framework for process assessment and improvement. TGPM can be
instantiated to the process under consideration to generate the corresponding
instantiated process model. A process is assessed or improved in relation to
some GQM goals. These goals are transformed into a goal tree. For the
refinement of a goal, we must identify the products, processes associated with
the goal and the participants who are responsible for each such product and
processes, and further what resource the participants use or can use [Pulford
(1995)]. The instantiated process model in conjunction with the description of
the corresponding internal process architecture can be used to build the goal
tree in a systematic manner [Satpathy (2000)].

**3.1 Analysis of TGPM**

The distinctive feature of the TGPM is the dual perspective of a process
attribute. In plain terms, it says how good a process attribute is in relation to
the process itself and then how it is related to the output product(s) of the
process and also to the end-product. This duality point of view identifies
certain process features such as (a) process faults, (b) process
understandability, (c) process scalability, (d) process reliability, (e) process
stability etc [Satpathy (2000)]. These features were not properly addressed by
any of the previous process models. Our interviews with our industrial
collaborators concluded that the dual view of process attributes exposes
certain relevant process issues, and addressing them was extremely important.

**4 PROFES Methodology**

The PROFES improvement methodology identifies a few important quality
goals of the final product which need to be improved. Typical such goals are:
(a) decrease the defect density of the final product, (b) higher reliability of the end-product, (c) predictability of the quality, time and cost of the development of the product etc. Next, all improvement actions are identified from a PPD (Product-Process Dependency) repository, the implementations of which, possibly, will lead to the desired product quality improvements; and a few important ones are identified and implemented.

PPD repositories are the core element of the PROFES improvement methodology. They contain an organized collection of PPD models. A PPD model describes the impact of a particular improvement action on a certain software quality characteristic when applied in a certain development process in a specific project context. Tables 3 shows the structure of a PPD model that we have used [Pfahl (2000)]. It says that, to get a better defect density in the final product, formal inspection should be applied to code development process under the following context characteristics. That the project type should be either semi-detached or embedded; it can be of any project size, and the manpower skill must be either average or high.

<table>
<thead>
<tr>
<th>PPD Model A</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Quality</td>
<td>Defect Density</td>
</tr>
<tr>
<td>Process</td>
<td>Code Development</td>
</tr>
<tr>
<td>Improvement action</td>
<td>Formal Inspection</td>
</tr>
<tr>
<td>Context Section</td>
<td></td>
</tr>
<tr>
<td>CF-1</td>
<td>Project Type</td>
</tr>
<tr>
<td>CF-2</td>
<td>Project Size</td>
</tr>
<tr>
<td>CF-3</td>
<td>Manpower Skill</td>
</tr>
</tbody>
</table>

Table 3. PPD model A.

4.1 Evolution of a PPD Repository

An organization is expected to have its own PPD repository for its PFPI programme. Initially, the organization should start with a tentative list of PPD models based on textbooks, experience reports, or can be obtained from the PROFES PPD repository [PPD (1999)]. It is obvious that such PPD models are generic in nature and they must be tailored for a specific organization. Such refinement of PPD models can be performed by using past project data and/or by conducting process assessment(s). PFPI can then be initiated on the basis of the refined PPD models. Every improvement cycle will come up with new knowledge, which, in turn, can be used to refine the corresponding PPD models. Thus, over a period of time, the customized PPD repository of an organization will attain a level of stability.
5 Comparison of TGPM and PROFES methodology

1. The TGPM and the PROFES methodology both aim at making the relationship between a process and its impact on product quality explicit. The instantiated process models can be enriched through measurement programmes over various runs of the process. From such enriched process models, it appears that the PPD models can also be generated; in other words, the enriched models will have most of the information required by the PPD models. Of course PPD models are better organized. But the instantiated models may provide certain improvement suggestions which can have impact on product quality, which PPD models may not provide. We will discuss this issue in our case study.

2. Let us analyse the structure of a PPD model (refer to Tables 3). Its product quality factor is defined in relation to the ISO 9126 model. The associated information on process and improvement action is usually obtained from experience. And the context factors of a PPD model in a sense customize the original generic PPD model. In the case of the TGPM, the ISO 9126 model is embedded in it. So the relationship of a process to a product is usually defined in relation to the ISO 9126. Further, the application context parameter during the process of instantiation of the TGPM resembles the refinement of PPD models through context factors.

3. The TGPM is defined over a formalized internal process model; i.e., every instantiated TGPM has an associated internal process model. The rationale behind a PPD model may become far more clear when seen from the viewpoint of the internal process architecture of the TGPM. Otherwise, it will be seen as a black-box rule.

4. A measurement program is a key technology in the PROFES methodology. Usually, the GQM method is followed to implement such a measurement program. Fuggetta et al. [Fuggetta (1998)], based on their experience in applying the GQM method at Digital Laboratories, have observed that the development of the GQM and measurement plans must be based on a comprehensive knowledge of the process details. Such knowledge decreases the semantic gap between the high level GQM goals and the questions that need to be asked in building up the GQM tree. Therefore, the PROFES methodology needs to have comprehensive knowledge of processes. The internal process model of TGPM can provide the PROFES methodology with such comprehensive knowledge.

5. The PROFES methodology aims at improving the product quality of the final product. However, there are many intermediate products that may affect product quality. Some examples could be intermediate products like a formal specification or formal design. The instantiated models of TGPM address such issues in an elegant manner.
6. TGPM and initial PPD repository of an organization are both generic in nature. TGPM is instantiated to various processes with the knowledge of the specific organization and application contexts. The generic PPDs are also adapted for an organization by the use of the knowledge of the organization and the type of projects the organization handles.

7. Process changes could be revolutionary; for instance, an organization rejecting the SSAD method and choosing instead an OOD method. In such a case, its PPD repository may suddenly become unstable. However, in such a case the instantiations of TGPM could proceed with ease. And then the new PPD models could be obtained (which may be tentative in nature) from the instantiated models.

6 A Case Study

We have performed a case study for comparing formal and informal specifications [Satpathy (2001)]. The object of our case study is the teletext module of a new generation TV from Philips Electronics. We have specified this module both formally in B [Abrial (1996)] and informally in UML [Rumbaugh (1999)]. Figure 2 presents an overview of the teletext system. Teletext pages are transmitted over the transmission channel, and a user can display them by pressing the keys on the TV remote. The sizes and the effort log for both the specifications are as given respectively in Tables 5 and 6.

![Figure 2. The Teletext System.](image)

A few comparative studies have been done to compare and contrast formal and informal specifications [Draper (1996), Larsen (1995), Snook (2001)]. The general observation is that formal specifications do not require more effort than corresponding informal specifications. In our case study, since the formal specification process took more time (17% more), we tried to see if the cycle time of the formal specification could be reduced.
<table>
<thead>
<tr>
<th>Specification Language</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>14 B Machines</td>
</tr>
<tr>
<td></td>
<td>48 pages of specification</td>
</tr>
<tr>
<td>UML</td>
<td>18 UML diagrams</td>
</tr>
<tr>
<td></td>
<td>52 pages of documentation</td>
</tr>
</tbody>
</table>

Table 4. Sizes of the UML and B Specifications.

<table>
<thead>
<tr>
<th>Specification Language</th>
<th>Effort (in Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>221</td>
</tr>
<tr>
<td>UML</td>
<td>183</td>
</tr>
</tbody>
</table>

Table 5. Effort log for obtaining the B and UML specifications

In order to evaluate our specifications, we instantiated the TGPM to the FS process and informal specification process. Following the GQM method, we took measurements of the metrics for important quality attributes in relation to the instantiated models. We then determined that the process attributes like process reliability, cycle time estimation, informal verifiability etc. which could have impact on the cycle time. Analysing the above attributes (process reliability and informal variability to be specific), we noticed that a lot of time has been spent on animation of B machines, the reason being problems with tool support for B. Our metric framework (based on TGPM) recorded these problems as process failures under the heading Process Reliability. Further, the effort log for Informal Verifiability was high because of the problems with the animator. The suggested improvement action was to go for an improved tool support. However, we have not implemented such an improvement action since we were using an expensive tool, i.e. the B-Toolkit. We have informed our feedback to the tool developers and the B-Toolkit is being upgraded based on our feedback [B-Core (2000)]. Note that the present case study was performed in an academic environment, and therefore we did not need to use PPD models in the study.

PPD models are usually tailored for the end product quality but not for the quality of intermediate products. Even if PPD models are extended for intermediate products, our observation is that no other quality model has attributes like process faults and process defects. So, probably, such PPD models will miss the improvement actions we discovered because of our dual interpretation of process attributes in TGPM. So, we claim that TGPM will play a good supportive role in obtaining PPD models under the PROFES methodology.
7 Conclusion

In this paper, we have discussed the TGPM and the PROFES methodology, and shown how there can be cases when TGPM can supplement the PROFES methodology. The internal process model architecture in case of TGPM can provide a better context of a PPD model to the people involved in the improvement programme. Further, since the TGPM emphasizes the intermediate products as well as the final product, it can suggest better improvement actions than the PPD repository. We have demonstrated this through a case study.

References

B-Core. (2000). B-Toolkit, Release 5.0.6, B-Core(UK).


**Appendix A: Definitions of some subfactors of the TGPM**

**Consistency:**
(i) The degree to which the process uncovers contradictions in the input product set.
(ii) The degree to which the process does not introduce contradictions in the output product set.

**Defect Trend:**
(i) The trend of defects that are observed in the process itself (defects linked to the process – during or after process execution).
(ii) The degree to which the process detects the defects or deficiencies in the product set under transformation so that defects in the output product set and the final product are minimal.

**Appendix B: subfactors after instantiation to the FS Process**

**Consistency:**
(i) Degree to which the FS process uncovers contradictions in the RD.
(ii) Degree to which the FS process does not introduce contradictions in the FS.

**Defect Trend:**
(i) The trend of defects that are observed in the process itself (defects linked to the process – during or after process execution).
(ii) The degree to which the process detects defects or deficiencies in the RD so that defects in the FS and the final product are minimal.