Investigation of Product Process Dependency Models through Probabilistic Modeling

Manoranjan Satpathy Rachel Harrison Daniel Rodriguez

School of Computer Science, Cybernetics and Electronic Engineering
University of Reading, Reading RG6 6AY, UK
{M.Satpathy, Rachel.Harrison, d.rodriguez-garcia}@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. A PPD model tells which improvement action will result in the improvement of which product quality. Because of the associated cost, only a small number of improvement actions can be implemented. Therefore it becomes imperative that we must be sure of the impact of the improvement actions. In this paper, we discuss how Bayesian Networks [Jensen (1996)] can be used to predict the outcome of PPD models and hence the impact of the associated improvement actions.

Keywords: *Process Improvement; PROFES Methodology, Bayesian Networks*

1 Introduction

Continuous process improvement is integral to all 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. Such processes mostly fall into the categories of primary and supporting processes of the ISO 12207 [ISO (1995)] process model. 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; in other words, many improvement actions are possible for any individual process. However, 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)].

The organization of the paper is as follows. Section 2 discusses the PROFES methodology. Section 3 discusses the prediction of the outcome of PPD through Bayesian networks. Section 4 describes a case study and Section 5 concludes the paper.

2. PROFES Methodology

The PROFES improvement methodology identifies a few important quality goals of the final product which need to be improved. Typical goals are: (a) decrease the defect density of the final product, (b) increase reliability of the end-product, (c) increase fitness for use of the final product, (d) improve predictability of the quality, time and cost of the development of the product etc. Next, all improvement actions are identified from a PPD repository, the implementations of which, possibly, will lead to the desired product quality improvements. Of course, the number of such improvement actions could be many, in which case the 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. Table 1 shows the structure of a PPD model that we have used [Pfahl (2000)].

PPD model A 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 personnel manpower skill must be either average or high.

PPD Model A		
Product Quality		Defect Density
Process		Code Development
Improvement action		Formal Inspection
Context Section		
CF-1	*Project Type*	*Organic,**semi-detached,embedded***
CF-2	*Project Size*	***Small,average,large***
CF-3	*Manpower Skill*	*Small, **average, high***

Table 1. PPD model A.

The six phases of PROFES improvement cycle are as follows [van Latum (2000)].

- **Characterize:** Current product quality is evaluated by analyzing available product quality data. Customer feedback, customer surveys, market research results, internal interviews etc. provide information for product quality needs.
- **Set Goals:** Final product improvement goals are set. Necessary process changes are determined by referring to the PPD repository. Those PPDs which match the current project context and the product quality needs are the candidates for selection. Corresponding process improvement actions are then obtained. If improvement actions are many, then the most important ones are selected by expert judgment or from previous simulation results [Pfahl (2000)].
- **Plan:** Plans for implementing the process changes are made which may include training needs, progress training etc. A measurement program, possibly based on the GQM method [Basili (1994)], is initiated.

- **Execute:** The improvement actions are implemented according to the plan. If necessary, corrective actions based on feedback analysis are taken.
- **Analyse:** Evaluate the measured data to find out if the improvement actions indeed resulted in improved product quality. The lessons learned are captured and evaluated.
- **Package:** Experience gained from the project is stored for further use. The PPD models in the repository may be enriched based on the acquired knowledge.

2.1. Evolution of a PPD Repository

An organization is expected to have its own PPD repository for its PFPI programme. But questions may arise as to how an organization should go about it? The organization should start with a tentative list of PPD models based on textbooks, experience reports, or obtained from the PROFES PPD repository [PPD (1999)]. The PROFES project has identified a set of such models from an extensive investigation of the process-impact on product quality. 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.

3. PPD model validation

Under the PROFES improvement methodology, it is crucial that the PPD models be valid. One way to validate PPD models is to generate empirical evidence from pilot applications. However, such an approach is not only expensive but it also can be risky. Therefore, it would be appropriate to use simulation models to predict the outcome of PPD models before they are selected for implementation. Pfahl et al [Pfahl (2000)] have used system dynamics to perform such a simulation of PPD models. In this paper, we will use Bayesian networks.

3.1. Bayesian Networks

A Bayesian Belief Network (BBN) [Jensen (1996), Fenton (1999)] is a directed graph in which the nodes represent uncertain variables and the arcs represent the causal relationship between the variables. Each node has a probability table, which stores the conditional probabilities for each possible state of the node variable in relation to each combination of its parent state values. For a node without any parents, such a table stores the marginal probabilities for each possible state of that node. If the state of a certain node is known then its probability table is altered to reflect this knowledge. Such knowledge is then propagated to determine the changed probabilities of all possible values associated with other nodes. Note that the initial probabilities of the nodes in a BBN are obtained from expert judgment and past project data. In fact, tools are available to help in the generation of BBNs from historical project data [Pronel (2001)].

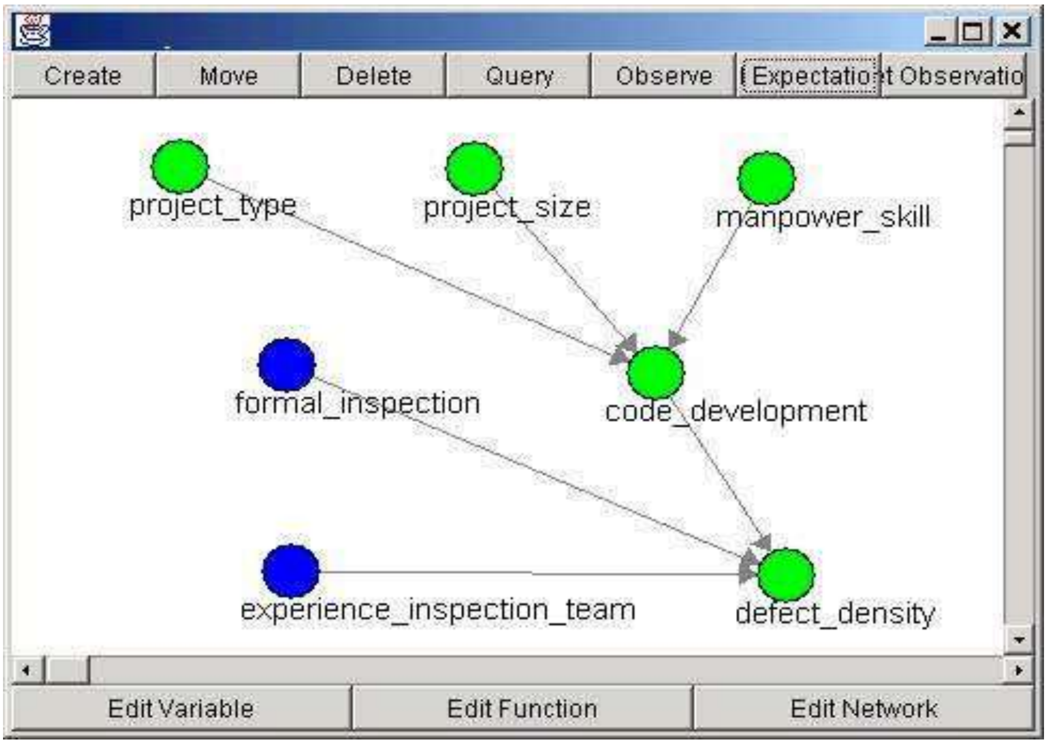

Figure 1. BBN for PPD model A.

The PPD model of Table 1 can be represented as a BBN (see Figure 1). The BBN shows that the quality of code development depends on project-type, project-size and manpower-skill. The defect density in the final product depends on formal inspection and the experience of the inspection team. Conditional probabilities are assigned to each node through expert judgment. Figure 2 shows that when there is no formal inspection and the quality of developed code is medium, the defect density of the final product is likely to be high. However, if we use an experienced inspection team, then the defect density can be brought down. Such a scenario is shown in Figure 3.

The probabilities associated a BBN representing a PPD model are supposed to be tuned to the capability of an organization; in other words, such values represent to what extent each of the context factors influence the outcome of the application of the PPD model. Such values are refined as new experience is gathered from implementation of improvement actions. So when we used a network to predict the outcome of the corresponding PPD model in advance, in all likelihood, we would succeed in our prediction. Coming back to the example discussed above, if a number of PPDs are available for decreasing defect density then their corresponding BBNs can be analysed, and such a analysis will reveal which PPD model is most effective in bringing defect density down. Further, the impact of a combination of PPD models can also be obtained by constructing a BBN which can predict the combined impact of the corresponding improvement actions. The appropriate combination can then be selected for implementation.

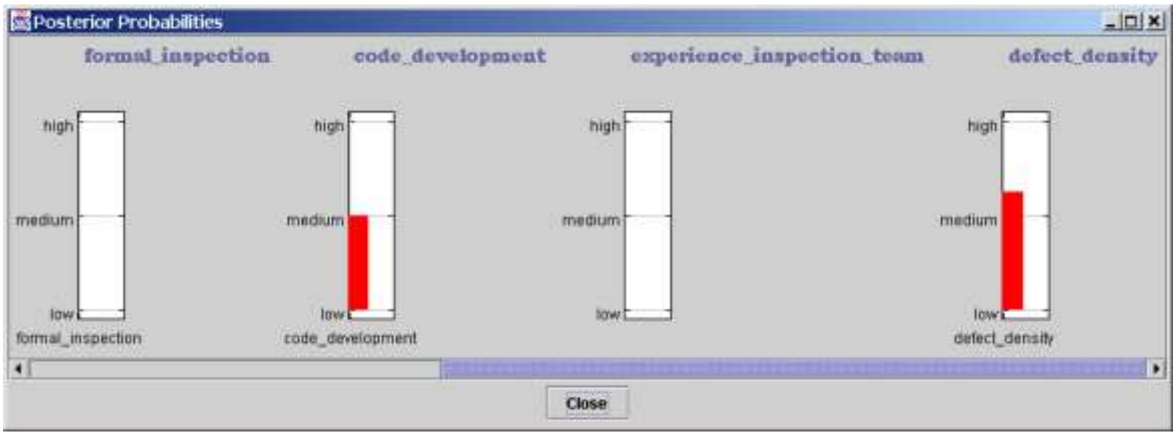

Figure 2 Analysis of BBN without formal inspection.

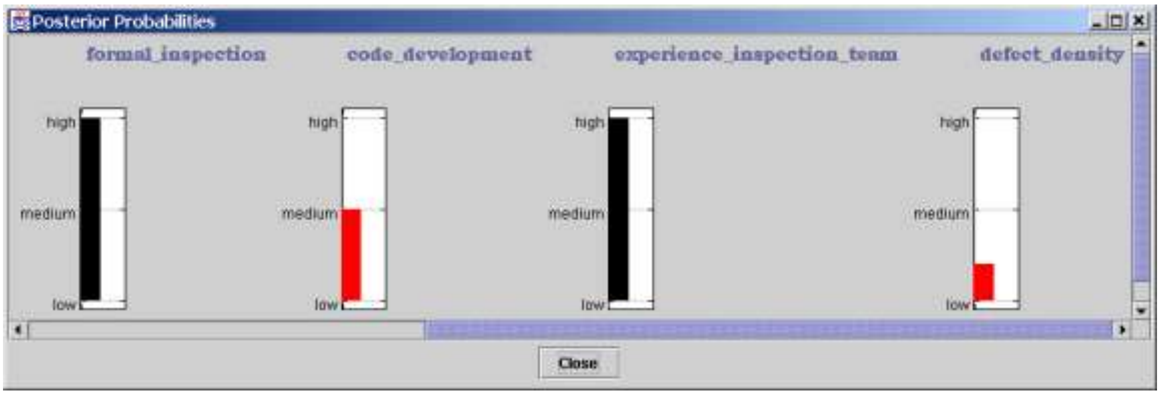

Figure 3 BBN analysis with an experienced formal inspection team

4. 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)] and compared both specifications. Figure 4 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.

A few comparative studies have been done to compare and contrast formal and informal specifications [Draper (1996), Larsen (1996), 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) in relation to the time that it took to generate the informal specifications, we tried to see if the cycle time of the formal specification could be reduced.

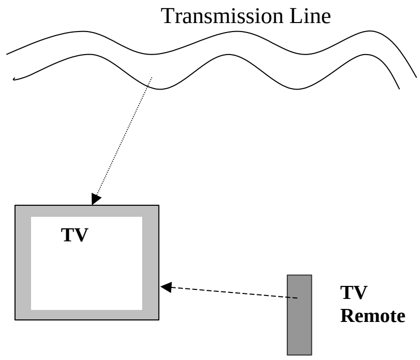

Figure 2. The Teletext System.

PPD models are usually used to improve the quality of the final product. In our case study, we have generated PPD models assuming that formal inspection is the final product; in short, we have extended the applicability of PPD models to important intermediate products. Our BBN analysis suggested improving tool support in order to reduce the cycle time of the generation of formal specifications. The measurement data of the case study also revealed that a significant portion of the time was wasted because of bugs in the tool. We have not implemented such an improvement action since we were using an expensive tool. The tool is being upgraded based on our feedback [Sorensen (2000)]

5. Conclusion

In this paper, we have discussed how predicting the outcome of PPD models before they are implemented is important. We have discussed how PPD models can be better represented as BBNs and analysis on them can predict the impact of the PPD models. The impact of a combination of PPDs on the product quality can also be analysed through BBNs. We have used BBNs to predict the outcome of some PPD models in our case study.

Acknowledgements: The authors wish to acknowledge the support of UK EPSRC which has funded EMPAF (ER/L87347) and that of CICYT TIC2001-1143-C03 (Spain) which has funded the ARGO project.

References

Abrial J.-R. (1996). The B—Book: Assigning Programs to Meanings, Cambridge University Press.

Basili V.R., Caldiera G., Rombach H.D. (1994). The Goal Question Metric Approach, The Encyclopaedia of Software Engineering, John Wiley.

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

Draper, J., Treharne, H., Boyce T., and Ormsby, B. (1996). Evaluating the B-Method on an Avionics Example, Data Systems in Aerospace (DASIA) Conference, Rome, June 1996 pp 89-97 (European Space Agency Publication Division WPP-116).

Fenton N.E., Neil M. (2000). Software Metrics: Roadmap, invited paper for ICSE 2000..

Humphrey W.S. (1992). Introduction to Software Process Improvement, SEI Report CMU/SEI-92-TR-7.

ISO (1991). International Standard: ISO/IEC 9126, ISO, Geneva.

ISO (1994). ISO 9001, Quality Systems – Model for Quality Assurance in Design, Development, Production, Installation and Servicing, 2nd Ed., ISO.

ISO (1995). ISO/IEC 12207, Software Life Cycle Processes, 1st Ed., Aug 1995.

Jarvinen J., Komi-Sirvio S., Ruhe G. (2000). The PROFES Improvement Methodology – Enabling Technologies and Methodology Design *in Product Focused Software Process Improvement (Eds. F. Bomarius and M. Ovio)*, Proc. of PROFES 2000, Oulu, Finland, LNCS volume No. 1840, Springer.

Jensen F.V. (1996) An Introduction to Bayesian Networks, UCL Press.

van Latum F., van Uijtregt (2000). Product Driven Process Improvement: PROFES Experiences at Drager *in Product Focused Software Process Improvement (Eds. F. Bomarius and M. Ovio)*, Proc. of PROFES 2000, Oulu, Finland, LNCS volume No. 1840, Springer.

Larsen, G.M., Fitzgerald J., and Brookes, T. (1996). Applying Formal Specification in Industry, IEEE Software, May 1996.

Pfahl D., Birk A. (2000). Using simulation to visualize and analyse product-process dependencies in software development projects *in Product Focused Software Process Improvement (Eds. F. Bomarius and M. Ovio)*, Proc. of PROFES 2000, Oulu, Finland, LNCS volume No. 1840, Springer.

Pronel. (2001). Website *http://www.tiga-technologies.com/projets/pronel/pronel.html*

PROFES (2001). The PROFES Methodology, website http://www.profes.org.

PPD (1999). The PROFES PPD Repository: Understanding patterns of Product/Process Dependence, website: http://www.iese.fhg.de/projects/profes/PPDRepository.

Rumbaugh J., Jacobson I., Booch G. (1999). The Unified Modeling Language Reference Manual, Addison Wesley.

Satpathy M., Harrison R., Snook C., Butler M. (2001). A Comparative Study of Formal and Informal Specifications through an Industrial Case Study, Proc. of the IEEE/IFIP Joint work shop on Formal Specification of Computer based Systems (Eds. C. Rattray, M. Sveda, J. Rozenblit), Washington DC, April 2001.

Sorensen I. (2001). *Personal communication*.

Snook C.F., Harrison R. (2001). Practitioners' Views on the Use of Formal Methods: An Industrial Survey Structured Interview, Information and Software Technology, March 2001.