PROJECT RISK MANAGEMENT BASED ON BAYES BELIEF NET USING EMF

Ákos SZŐKE
Advisor: András PATARICZA

I. Introduction

Although software metrics is nearly 30 years old it could not become a silver bullet to project management (PM). The key reason is that metrics does not provide information to the main objective of PM: decision support during software development [1]. Some exhaustive investigation [2,3,4] concluded that Bayes Belief Nets (BBNs) provide far the best solution to this managerial problem. This paper describes an extensible and powerful model based risk management tool which intends to help governing the development of today's complex software intensive systems.

The present investigation is related to a GVOP tender (GVOP-3.3.3.-05).

II. Quantitative Risk Management Based on Bayes Belief Net

In everyday project management, decisions are based on mostly subjective information, which is usually uncertain and incomplete. Adapting agile development methodologies such as Extreme Programming (XP) require even more precise and robust decision-making, which implies support for risk management in an easy way. In mature risk management, the decisions should be based on quantitative information which can be gained through the widespread CMU SEI PM level software metrics [5]. Unfortunately, metrics based approaches do not support decision-making; they just provide some useful data for risk analysis.

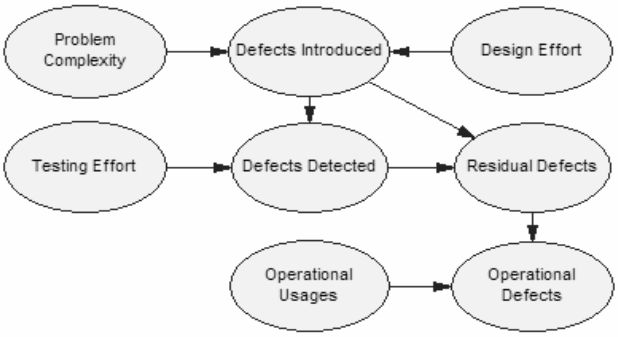

Figure 1: A defect model expressed by BBN

BBNs have proven to be an extremely powerful technique for reasoning under uncertainty. A BBN is a directed acyclic graph (DAC), where nodes of a BBN represent uncertain variables and the arcs are the causal links among them. Between each node, a set of conditional probability functions (CPF) are defined to model the uncertain relationships.

A defect model [3] expressed by BBN can be seen in Figure 1. Like any BBN, this model contains a mixture of variables where some values are known, and others are interested. The power of BBN is that it will compute the probability of every variable irrespectively of the amount of known variables to support decision-making.

III. Modeling BBNs with EMF

EMF (Eclipse Modeling Framework), a core part of the Eclipse IDE, is an implementation of OMG's MOF [6], and besides it is a Java framework and code generation facility for building tools based on structured data models. These models are simply a set of related classes used to handle the data in an application.

One of the main components of EMF is ECore framework which is responsible for basic model generation. Although, ECore is a model, it is a metamodel, or meta-metamodel depending on how it

is used. In this presented approach, BBN models are defined with ECore in multiple abstracted way: ECore is a meta-metamodel (M3 model) because it is used to define BBN metamodel (M2 model), and this metamodel defines how BBN models (M1 model) can be created.

IV. Overview of the Model-driven BBN-based Decision Support

The proposed solution consists of an Eclipse plug-in for BBN (meta)modeling and a popular BBN engine (Hugin Decision Engine 6.6) [8] for probability computations and BBN updates.

A BBN model describes managerial risk management problem to be supported. In the solution architecture its position can be seen in Figure 2. Hence the BBN engine can import BBNs in its own structured language and the BBN model is in XMI (XML Metadata Interchange) format, an XMI to BBN engine language transformation is needed. This is realized by a very powerful tool for generating source code: JET (Java Emitter Template).

In order to create and edit a BBN model, a BBN metamodel should be composed with EMF. This metamodel (*.ecore) consists of a package, a few data types, enumerations, and classes, which are in fact instances of model elements in ECore.

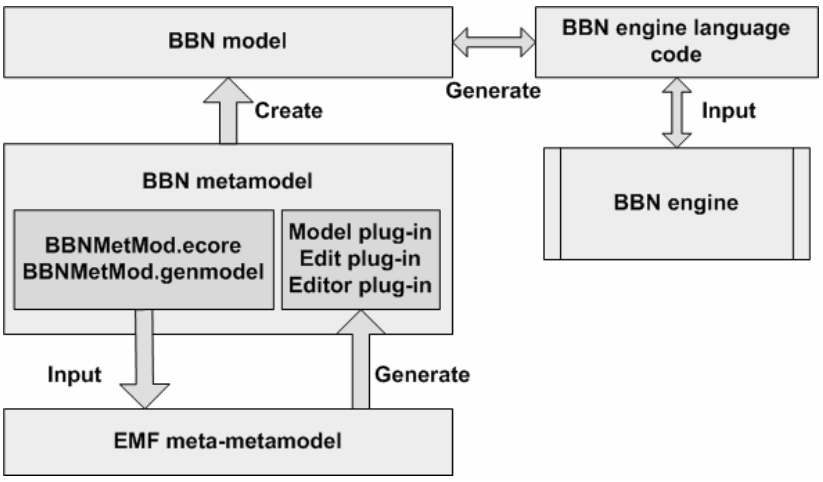

Figure 2: Solution architecture

From this metamodel, with the ECore's EMF.edit framework a generator model (*.genmodel) generates the desired Eclipse editor plug-in, as an Eclipse platform standard extension, for BBN creating and editing.

This integrated solution provides a flexible and extensible solution for decision-making.

V. Conclusion

The commonly used regression models may lead to inappropriate risk management decisions. The presented solution provides an extensible and powerful predictive model, where metrics are incorporated into cause-effect relationships (expressed by BBNs), to provide accurate predictions.

Our next objectives are to facilitate automatic BBN node set up with information gained through PM level software metrics and extend functionality with graphical editing capability using Eclipse's Graphical Editing Framework (GEF).

References

[1] V.R. Basili, H.D. Rombach, *Towards Improvement-oriented Software Environments*, IEEE TSE, p758-773, 1988

[2] N. E. Fenton, M. Neil, *A Critique of Software Defect Prediction Models*, IEEE Transactions on SE, p675-689, 1999

[3] N. E. Fenton et al, *Making Resource Decisions for Software Projects*, Proc. 26th ICSE, IEEE 2004

[4] C-F. Fan, Y-C Yu, *BBN-based Software Project Risk Management*, The Journal of Systems and Software, 2004

[5] Carnegie Melon University Software Engineering Institute, *Software Metrics Guide*, URL: http://sunset.usc.edu

[6] T. Aven, *Foundations of Risk Analysis: A Knowledge and Decision-Oriented Perspective*, John Wiley & Sons, 2003

[7] C. G. Wu, *Modeling Rule-Based Systems with EMF*, Eclipse Corner Article 2004

[8] B. Moore et al, *Eclipse Development using the GEF and the EMF*, IBM Redbooks 2004

[9] Hugin Expert A/S, *HUGIN API Reference Manual*, URL: http://www.hugin.com