Automated Population of Causal Models for Improved Software Risk Assessment

Peter Hearty
Queen Mary,
University of London, Mile End Road
London, England E1 4NS
+44(0) 207 882 7896
hearty@dcs.qmul.ac.uk

Norman Fenton and Martin Neil
Agena and Queen Mary,
University of London, Mile End Road
London, England E1 4NS
+44(0) 207 882 7860
norman@dcs.qmul.ac.uk

Patrick Cates
Agena Limited
32-33 Hatton Garden
London, England EC1N 8DL
+44 (0) 207 404 9722
patrick@agena.co.uk

ABSTRACT
Recent work in applying causal modeling (Bayesian networks) to software engineering has resulted in improved decision support systems for software project managers. Once the causal models are built there are commercial tools that can run them. However, data to populate the models is typically entered manually and this is an impediment to their more widespread use. Hence, here we present a prototype tool for automatically extracting a range of relevant software metrics from popular project management and CASE tools. This information is used to populate Bayesian networks with the aim of providing better real world predictions of the risks associated with software costs, timescales and reliability.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management – *cost estimation, time estimation, software quality assurance.*

D.2.8 [**Software Engineering**]: Metrics – *process metrics, product metrics*.

K.6.1 [**Management of Computing and Information Systems**]: Project and People Management – *management techniques*.

General Terms
Management, Measurement.

Keywords
Bayesian networks, software process models.

1. INTRODUCTION
The value of Bayesian Networks (BN) in software project management has recently been demonstrated [1, 2, 3, 10]. BNs have been constructed which successfully encapsulate results from empirical software engineering research in intuitive, easy to use models. For example, in [3] we described a model that has been used successfully for software project risk assessment that incorporated trade-offs between resources, schedule, quality and functionality; in [8] we described a model that has been used to achieve significantly improved defect prediction. In particular, extensive trials at Philips have shown a 95% accuracy in predicted defects (correlation between actual and predicted). This compares with previous best levels around 70% [9]. In addition to providing accurate predictions of useful project attributes, BN models provide estimates of the risk associated with each prediction [6].

An important step on the way to wider industry acceptance of these models is the integration of data and metric calculations from a wide variety of popular tools. We aim to demonstrate how such an integration might proceed and the benefits to be gained from doing so.

ASE'05, November 7–11, 2005, Long Beach, California, USA.
ACM 1-58113-993-4/05/0011.

2. BACKGROUND
Software project models built using BNs allow the integration of a large number of disparate software metrics and expert judgments into a single, intuitive, visual model. Integration of these many sources of data not only provides greater prediction accuracy but also widens the scope of the models to benefit a larger group of project stakeholders. The same model that generates estimates of timescales and resources also predicts quality attributes such as defect densities and mean time between failures.

However, collecting software process and product data is widely perceived to be a labor intensive task. Many managers see it as an unwanted overhead. This is particularly true where some of the data is used solely for baseline calibration purposes, its benefits only being apparent in later projects.

Not only is the overhead of data collection significant, the very act of entering data into a model is a time consuming and error prone task. The model must be continuously maintained, often requiring duplicate data entry into other management tools and software management systems.

The ideal tool would deliver the benefits of improved BN software project modeling without incurring the additional data collection and data-entry overheads. What is needed is better integration between management models and other management and development tools.

3. INTEGRATED TOOLS
Much of the data required by the BN project models can already be found in existing management and development tools. For example:

1. Project management tools contain task size and resource estimates.

2. CASE tools contain object, module and data decompositions and relationships.
3. IDE and code management tools contain code size and complexity measures.
4. Bug tracking tools provide information on defects and failure rates.

Automated extraction of data from the above tools allows BN project models to be updated instantly and transparently with little or no effort on the part of managers, developers or testing personnel.

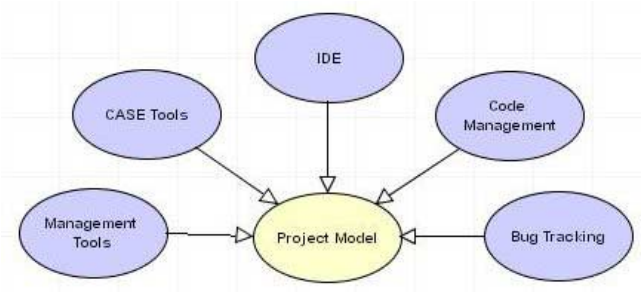

Figure 1. Data integration from multiple tools.

The AgenaRisk database add-in accepts data via any JDBC compliant database driver. A wide range of both relational and non-relational data sources are therefore available. The database add-in allows named, parameterized queries to be defined against each data source. The output of any query can be used to supply parameters to multiple instances of child queries, creating a hierarchy of dependent queries and data sets, spread across multiple data sources.

By defining network fragments as BN classes, query results can be used to instantiate chains of BN objects with customized node properties, allowing large, complex models to be constructed with relative ease.

Multiple scenarios can be run by specifying a single query and automatically invoking "child" queries parameterized using the output of a parent query. Model parameters can therefore be collected for sensitivity analysis or learned via appropriate regression methods.

4. RELATED RESEARCH

Large, realistic, models based on BNs have been possible since the discovery of efficient BN implementation algorithms [7]. This led Fenton and Neil [4] to propose the use of BNs to model software defect prediction. A series of ever more sophisticated models followed, culminating in the AID tool [8] the MODIST project [3], and the extensive trials of revised models in AgenaRisk at Philips [9].

These models will evolve to include database queries as separate nodes within the model. Some of the mathematical foundation for this work is provided in [5].

5. ACKNOWLEDGMENTS

This research is taking place as part of the eXdecide project, funded by the UK Engineering and Physical Sciences Research Council (EPSRC) and Agena Limited.

6. REFERENCES

[1] Bibi, S. and Stamelos, I. Software Process Modeling with Bayesian Belief Networks. In *Proceedings of 10th International Software Metrics Symposium (Metrics 2004)* 14-16 September 2004, Chicago, USA.

[2] Fan, Chin-Feng, Yu, Yuan-Chang. BBN-based software project risk management, *J Systems Software*, 73, 193-203, 2004.

[3] Fenton, N. E., Marsh, W., Neil, M., Cates, P., Forey, S. and Tailor, T. Making Resource Decisions for Software Projects. In *Proceedings of 26th International Conference on Software Engineering (ICSE 2004), (Edinburgh, United Kingdom, May 2004)* IEEE Computer Society 2004, ISBN 0-7695-2163-0, 397-406

[4] Fenton, N. E. and Neil, M. A Critique of Software Defect Prediction Models, *IEEE Transactions on Software Engineering*, 25(4):675-689, September 1999.

[5] Friedman, N., Getoor, L., Koller, D. and Pfeffer, A. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)* (1300-1307), 1999.

[6] Jensen, F. V. *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York, NY, 2001.

[7] Lauritzen, S. L. and Spiegelhalter, D. J. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J.R. Statistical Soc.* Series B, 50, no. 2, pp. 157-224, 1988

[8] Neil, M., Krause, P., Fenton, N. E., *Software Quality Prediction Using Bayesian Networks in Software Engineering with Computational Intelligence*, (Ed Khoshgoftaar TM), Kluwer, ISBN 1-4020-7427-1, Chapter 6, 2003

[9] Neil, M. and Fenton P. Improved Software Defect Prediction. *10th European SEPG, London, 2005*.

[10] Stamelosa, I., Angelisa, L., Dimoua, P., Sakellaris, P. On the use of Bayesian belief networks for the prediction of software productivity. *Information and Software Tech*, 45 (1), 51-60, 2003.