18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

An approach to rapid prototyping for a web-based risk management system

Y.B. Khoo [1], **M. Zhou** [1] **and B. Kayis** [2]

[1] Commonwealth Scientific and Industrial Research Organisation (CSIRO)
Email: yongbing.khoo@csiro.au
[2] University of New South Wales (UNSW)
Email: b.kayis@unsw.edu.au

Abstract: One of the most significant complexities of managing concurrent engineering (CE) projects is the identification and mitigation of project risks. The challenges that project managers encounter are compounded by constantly evolving business requirements and the large volumes of data involved in such projects. In this research, rapid prototyping techniques were used to accelerate the process of developing prototypes of a web-based risk management system to address such challenges.

Based on the concept of Model-Driven Architecture (MDA), InfoMAP (Information Modeling and Application Prototyping) was developed as an Eclipse Modeling Framework (EMF)-based Rich Client Platform (RCP) application to encapsulate relationships between information and system components. In particular, relationships and properties of risk types were parameterized to facilitate quantitative and qualitative refinements. InfoMAP models are made up of maps, which consist of alternating actions and entities (Zhou, 1996). The risk management system used these models to generate data structures and template-based user interfaces at run-time.

There are four main models in the system: context establishment, risk identification, risk analysis, and risk monitoring. The context establishment model helps users to identify systemic problems and adopt consistent mitigation strategies through the use of shared risk profiles and management priorities. Based on this contextual information, the risk identification model provides a structured process that guides users in responding to questions that are used to determine the level of risks in projects. Using the inputs to the previous two models, the risk analysis model is used to generate suggested risk likelihood and consequence values based on historical data from previous projects. The system uses the risk monitor model to extract significant risks for users to monitor and implement mitigation strategies. The significance of risks is customized according to each project and is set by users in the context establishment model.

The availability of early prototypes of the system allowed users to receive regular updates to the software and provide timely feedback. This iterative user engagement process was identified as one of the key factors that contributed to the success of the risk management system. The final production system was optimized by transforming the models into platform-specific software modules.

The analysis results of the risk management system were evaluated and used to revise its accuracy in the derivation of risk levels using actual costing data from past projects.

Keywords: *Rapid prototyping, Risk management, Web-based, Java, J2EE, EMF*

1. INTRODUCTION

Concurrent engineering (CE) is defined as "a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements" (Winner et al., 1988). Studies of the complexities of iterative and continuous risk management processes in CE projects have found that there is limited information available through knowledge elicitation (Lutters et al., 2001; Ong et al., 2006; Rozenfeld and Eversheim, 2002; Thannhuber et al., 2001). Because the effectiveness of CE may be determined by the degree of data sharing and reuse as well as the available support for decision-making in projects (Danesh and Jin, 2001), such limitations on information availability can prevent CE from reaching its full potential. In particular, aerospace engineering projects which typically span decades (to support the lifetime of most aircrafts) can derive huge benefits from the sharing and reuse of historical data extracted from past projects, so that project managers can adopt consistent and successful risk mitigation strategies. Without the sharing of existing knowledge, implemented mitigation strategies are usually based on managers' personal knowledge and perceived risk levels. The success rate of such decisions was found to be around 55.6% (Han and Diekmann, 2001).

This paper presents an approach to encapsulating complex relationships between risk types in CE projects based on information from past projects. This modeling paradigm was designed to facilitate the rapid prototyping of a web-based risk management system, establish project context, identify relevant risk types, analyze risk levels by including historical data, as well as support the interaction of risk types between different phases of CE projects. This provides a systematic approach to the quantification of potential risks at all stages of the project life cycle. Past and present knowledge on risk events are shared through lessons learnt, case studies, best practice and expert knowledge (Kayis et al., 2006).

2. RESEARCH UNDERPINNINGS

2.1. Information modeling

A key success criterion for CE projects is the maintenance of effective communication between team members (Smith, 1998). In view of this, Smith (1998) suggested that "members are co-located within conversational distance of each other". However, the challenge of communicating effectively can be still compounded by "the difficulties in understanding the diverse knowledge and information generated from these distributed/multi-disciplinary design teams" (Chao et al., 2002). As a result, the development of clear models of CE projects is fundamental to the effective management of these projects. These models should encapsulate the relationships between relevant information "effectively so that it can be interpreted and used" at an appropriate level of *processable expressiveness*, "the degree to which a mechanism supports machine understanding or semantic interpretation" (Webster, 1988). The importance of information modeling may be demonstrated by the corpus of related work. The major drivers of modeling approaches include Unified Modeling Language (UML), Extensible Markup Language (XML), and Model-Driven Architecture (MDA).

UML is a "visual modeling language, composed of notations and textual components to express object-oriented system designs" (Grossman et al., 2005). It consists of graphical models (known as structure and behavior diagrams) that encapsulate business processes and the corresponding software designs. Structure diagrams are used to model time-independent system components, e.g. algorithm for deriving risk magnitudes. On the other hand, behavior diagrams describe dynamic system states over time, e.g. the path of system execution as a result of user interaction. Although the UML has become increasingly popular in object-oriented software development processes, common concerns about UML include "it is too big and complex, it is semantically imprecise, it is implemented in a non-standard manner, it has limited customizability, it has inadequate support for component-based development, and that it is unable to easily interchange model diagrams" (Grossman et al., 2005).

XML is an open standard for creating custom plain text-based data structures to facilitate data interoperability between software packages (Harold and Means, 2004; ISO, 2005). This is in contrast to conventional relational databases which stores data in vendor-specific binary format. XML was initially targeted at information exchange on the Internet and has become prevalent in modern software platforms. In particular, XML is increasingly being used as descriptor files instead of proprietary configuration files. For instance, web-based server applications such as JBoss and Apache Tomcat as well as standalone Eclipse-based Rich Client Platform (RCP) applications rely on XML to provide customizable software behaviors (e.g. handling of concurrent user transactions) and user interfaces (e.g. support for different languages).

Drawbacks of using XML’s tree-like structures include the difficulties associated with the modeling of information that is not properly nested (Sperberg-McQueen and Huitfeldt, 2000) and the larger storage capacity required for XML files compared to identical data stored in binary format (Williams et al., 2005).

MDA was designed to address four primary objectives: “portability, productivity, interoperability and reusability by means of architectural separation of concerns and through the complete development lifecycle, covering analysis and design, programming, testing, component assembly, along with coding and maintenance” (Fernández-Medina et al., 2009). MDA leverages modeling frameworks such as UML and XML to facilitate platform independence. This is done by constructing Computation Independent Models (CIMs) by business analysts which are further developed into Platform Independent models (PIMs) by software architects. These PIMs are implemented as Platform Specific Models (PSMs) by software programmers (Fernández-Medina et al., 2009; Paige et al., 2005). Eclipse Modeling Framework (EMF) is an implementation of major MDA specifications through which applications can be generated using minimal sets of Java classes or EMF’s Ecore models that describe the relationships between data structures. For instance, RCP applications can be developed by first creating Ecore models (which are PIMs) and transformed into generic RCP editors that can be further customized.

2.2. Information modeling and application prototyping

Based on the concept of MDA, InfoMAP (Information Modeling and Application Prototyping) was developed as an EMF-based RCP application that allows users to encapsulate relationships between information and system components as PIMs. Systems can use these models to generate PSMs, e.g. dynamically create data structures and user interfaces. In this research, InfoMAP models were implemented as EMF models to generate rapid prototypes of the risk management system through which users were able to see changes and give feedback quickly. This iterative user engagement process was identified as one of the key factors that contributed to the success of software that was prototyped using InfoMAP.

InfoMAP models are made up of maps. Each map is a series of alternating actions and entities (Zhou, 1996). The entry point to each map is uniquely identified by the Start Entity (Figure 1). Entities are used to encapsulate data while actions refer to software modules that contain business logic. Each action (represented by a rectangle) and entity (represented by a circle) may comprise detailed maps that are constructed using generic actions and entities. An application may be made up of several maps whose objects are linked by unique identifiers.

3. WEB-BASED RISK MANAGEMENT SYSTEM

Most commercially available off-the-shelf risk management tools for CE projects do not have a systematic “risk roadmap” to support the identification, encapsulation, as well as visualization of the causal relationship between risk factors and their accumulated/inherited impacts in CE projects (Kayis et al., 2007). Furthermore, these tools do not readily support the reuse of lessons learnt from previous projects. As a result, a web-based Intelligent Risk Mapping and Assessment System (IRMAS) was designed and developed to support collaborative decision-making through the effective sharing of information (Kayis et al., 2006). The system was implemented as a J2EE web application with a database that stores information such as the relative importance of key success factors and lessons learnt/best practices. The information was collected through multiple interviews with domain experts. The database contained 589 risk items for different project types, information on 4372 risk items, and 136 lessons learnt extracted through detailed analysis of past projects.

The risk management process modeled by IRMAS is based on the AS/NZS 4360 standard (Standards Australia and Standards New Zealand, 2004). Probability is used to quantify perceived risk levels. These probability values are collected by interviewing a panel of domain experts and using historical data captured in the form of lessons learnt and best practices. This allows the modeling of risk types using incomplete knowledge of the underlying domain.

Relationships between risk management processes were encapsulated by four main InfoMAP models: context establishment, risk identification, risk analysis, and risk monitor. IRMAS was initially deployed using the information models so that the system can evolve rapidly according to business requirements. However, early prototypes of the system that were developed in this manner ran too slowly for production environments. The low system performance may be caused by the dynamic creations of data structures at run-time. As such, these models were transformed into optimized software modules before deployment.

3.1. Model 1: Relationships of risk types

The management of every organization is responsible for addressing and mitigating risks that are inherent in the organization's business model. Such contextual risks have to be identified in order to reduce the occurrence of systemic problems and increase the adoption of consistent mitigation strategies. Model 1 encapsulates the information exchange between users and IRMAS to determine the contextual data of projects including organizational and user details, project objective, ownership, management support, regulatory requirements, project type, schedule cut-off dates, estimated project budget, mitigation budget, as well as compliance to government and regulatory standards. In particular, the following data are used in the subsequent risk identification and analysis models:

- *Project type*. Supported project types are build-to-print, derivative design, and new design. Each project type has a set of pre-defined risk types. For instance, build-to-print projects do not need to consider the risk types associated with the certification of new products because certification was already obtained when the product was initially designed and produced.
- *Project objectives*. These objectives modify the relevant risk types, e.g. the objective to reduce the time to market influences schedule risks.
- *Suppliers*. IRMAS uses profiles to populate levels of risk types automatically and consistently. These profiles allow managers to maintain consistent perspective of risks associated with known suppliers through the elimination of (possible) personal bias.
- *Risk threshold.* Risks whose magnitude is above this value are considered significant.

CE risks are classified into eight categories in IRMAS:

- *Schedule risks* are related to issues surrounding project milestones, task dependencies, lead times, and production planning.
- *Technical risks* are related to professional trade and manufacturing issues such as mechanical, industrial and applied sciences (e.g. quality assurance).
- *External risks* are related to issues surrounding stakeholders who are outside the organization (e.g. changes in customer requirements).
- *Organizational risks* are defined by organization structure, ownership, as well as leadership styles.
- *Communicational risks* refer to the ability to exchange information within the organization and establish common understandings (e.g. language barriers, cultural differences, and communication media).
- *Locational risks* are the challenges posed by physical distance between project members and partners (e.g. geographical areas, number of sites, team sizes).
- *Resource risks* are the availability of supplies, skills, and support (e.g. material, labor, equipment).
- *Financial risks* are challenges surrounding monetary income and expenditure (e.g. exposure to foreign currency exchange rates, inflation).

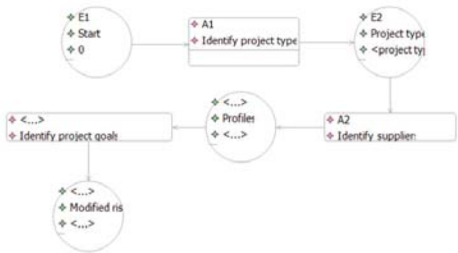

Figure 1. Part of the InfoMAP model used to establish.

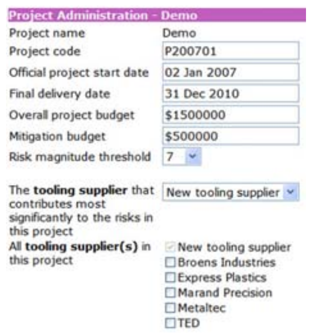

Figure 2. A screenshot of the webpage where project context is established.

These risk categories were derived from interviews with experienced project managers and team members using past projects as references. The collected information was encapsulated in InfoMAP models (Figure 1) and subsequently transformed into computational models within IRMAS. The resultant presentation is shown in Figure 2.

3.2. Model 2: Risk identification

Model 2 (Figure 3) is a conceptual model that represents a repository of questions that are classified according to the eight risk categories. The relevant questions and possible answers (determined using the project context that was established by Model 1) are stored persistently on a relational database. These questions are divided into six virtual phases in the project life cycle: conceptual design, preliminary design, detailed design, manufacture, certification, and customer support. Each of these phases contains risk types that can overlap and link to risk types of other phases.

Relevant information such as lessons learnt and examples (Figure 4) are derived by mapping the questions to the associated risk types. Such information can be used by project managers to better understand the risks in their own projects and provide appropriate responses to the questions.

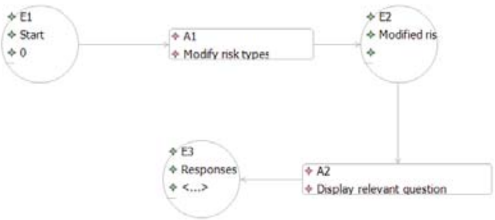

Figure 3. Part of the InfoMAP model used to encapsulate risk type questions.

3.3. Model 3: Risk analysis

In order to perform analyses of the risks identified by Model 2, IRMAS uses Model 3 (Figure 5) to encapsulate the relative importance of risk types and the probabilities of the occurrence of these risks. These probabilities are derived from the knowledge accrued in previous projects as well as a quantitative analysis of risks identified in case studies and lessons learnt.

The magnitude of each identified risk is derived from the product of the risk's likelihood and consequence. The likelihood of risk is defined as the relative frequency of risk event occurrence. It is computed using the Bayesian Belief Network (BBN) (Press, 1989) approach based on the probabilities encapsulated by Model 3. The consequence of risk is defined as the level of impact when the risk occurs. It is computed using the Analytical Hierarchy Process (AHP) (Saaty, 1999). Similarly, the AHP engine obtains inputs from the relative importance of risk types that are encapsulated in Model 3.

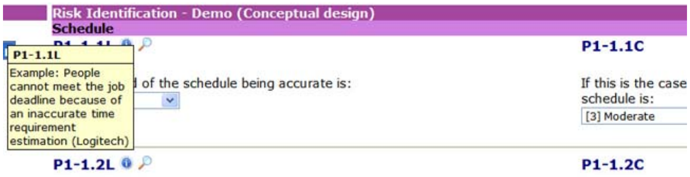

Figure 4. A screenshot of the risk identification page.

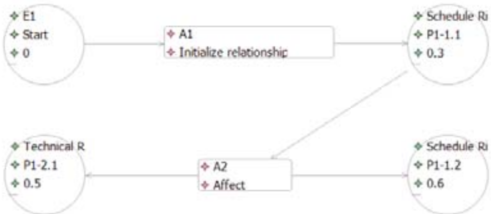

Figure 5. Part of the InfoMAP model that encapsulates the probabilities and relationships between risk types.

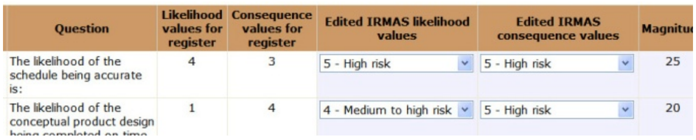

Figure 6. A screenshot of the risk analysis page.

For instance, the likelihood of making an inaccurate estimate of milestones may be rated as "up to 70%" while the consequence may be rated as "minor" because inaccuracies of the estimate are expected and already accounted for in the project schedule. These likelihood and consequence values are mapped to a five-point scale before the risk magnitude (1 to 25) is computed. Using users' responses for the current (and previous relevant) phase(s), IRMAS uses Model 2 to compute new sets of likelihood and consequence values for the risks in this phase. Using the first row of Figure 6 as an example, the computed values are both 5 (High risk). Validation tests of the computed values showed that the values are high because experience from previous projects indicated that schedules can be adversely influenced by factors that are beyond the organization's control (e.g. fluctuations in the foreign currency exchange). The analysis results (Figure 6) are used as inputs to Model 4.

3.4. Model 4: Risk register and other monitoring tools

Using the risk magnitude threshold identified in Model 1, Model 4 (Figure 7) extracts significant risks for users to monitor. This monitoring process involves the tracking of risk levels over time as a result of implemented mitigation strategies. Details of these strategies are stored in IRMAS's database (Figure 9) and can be added to the shared knowledge repository by authorized risk managers. The knowledge repository also contains information captured from generic engineering know-how, lessons learnt, case studies, best practices (external benchmarking) and engineering standards (e.g. AS/NZS 4360). Using a combination of Models 1, 2, and 4, users can search the knowledge repository for relevant lessons learnt based on the interactions between risk types and classification of the lessons learnt according to risk categories. Figure 8 shows a screenshot of the risk register page through which users can monitor changes in the risk magnitudes.

4. VALIDATION OF THE SYSTEM

Aberrant probability values in the risk analysis module (Model 3) were identified by querying the cumulative probabilities for risk types through the exclusion of user inputs. The ranges between these cumulative probabilities and user inputs were compared against the probability values derived by IRMAS's BBN engine. BBN models (which form parts of Model 3) of risk types that had BBN-generated values significantly outside these ranges were checked for data-entry errors. Furthermore, the outcomes of two previous large scale projects were used to validate the correctness of the BBN models. Results of this validation showed that the risk type probability model was 80% accurate (Figure 10) (Kayis et al., 2006).

Similarly, the costing data of previous projects were used to validate the risk magnitudes generated by IRMAS. The validation process involved comparing the ratio of the costing data to risk magnitudes. Risk types whose magnitudes were not proportional were revised by updating their probability values in Model 3 and the knowledge capturing process repeated using the Delphi technique (Rowe and Wright, 1999). The comparisons showed a good correlation between the risk magnitudes and costing data (Kayis et al., 2006).

5. CONCLUSION

This paper presented the use of EMF-based MDA approach to create InfoMAP models and generate prototypes of IRMAS, a web-based risk management system. IRMAS uses the models to structure users' risk management process through four steps: project context establishment, risk identification, risk analysis, and risk monitor. Details of new mitigation strategies are stored in the risk monitor step and relevant strategies can be incorporated into the shared knowledge repository by authorized risk managers for future reference.

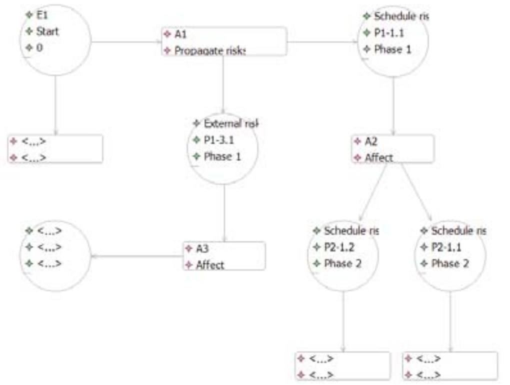

Figure 7. Part of the InfoMAP model that encapsulates the propagation effects of risk types to other phases.

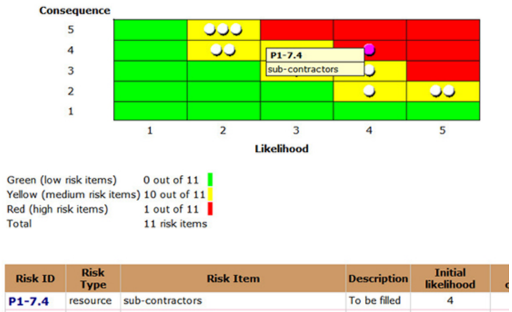

Figure 8. A screenshot of the risk register page that is used for monitoring the status of significant risk items.

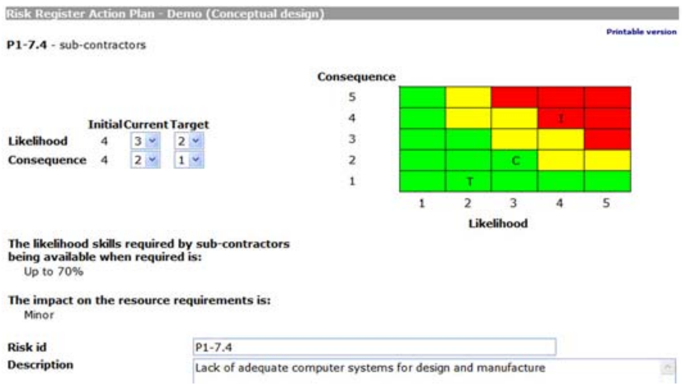

Figure 9. A screenshot of the risk register page where the details of mitigation strategies are stored.

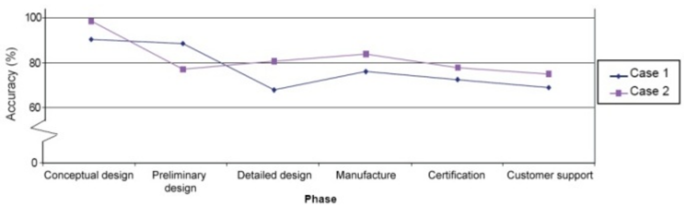

Figure 10. The accuracy levels of probability values of risk types using two case studies.

IRMAS prototypes were generated rapidly by embedding the models in a generic platform which created user interfaces based on pre-defined templates and constructed data structures at run-time. The relative ease through which these prototypes evolved to meet new user requirements allowed users to be engaged in early and regular validations of the system.

Future studies will concentrate on the customization of generic information models to support additional industry-specific requirements. These models may serve as a reference information model that can be used in areas such as workflow and cost analysis.

ACKNOWLEDGMENTS

This research was funded by the Cooperative Research Center for Intelligent Manufacturing Systems and Technologies (CRCIMST) Australia. We acknowledge the contributions of the IRMAS project team members, especially Drs. Sue Savci and Adrian Rispler from Boeing-Hawker de Havilland.

REFERENCES

Chao, K., Norman, P., Anane, R., and James, A. (2002), An agent-based approach to engineering design. *Computers in Industry*, 48(1), 17-27.

Danesh, M. and Jin, Y. (2001), An agent-based decision network for concurrent engineering design. *Concurrent Engineering*, 9(1), 37-47.

Fernández-Medina, E., Jurjens, J., Trujillo, J., and Jajodia, S. (2009), Model-driven development for secure information systems. *Information and Software Technology*, 51(5), 809-814.

Grossman, M., Aronson, J. E., and McCarthy, R. V. (2005), Does UML make the grade? Insights from the software development community. *Information and Software Technology*, 47(6), 383-397.

Han, S. H. and Diekmann, J. E. (2001), Approaches for making risk-based go/no-go decision for international projects. *Journal of Construction Engineering and Management*, 127(4), 300-308.

Harold, E. R. and Means, W. S. (2004), XML in a nutshell (3rd ed.). O'Reilly Media, Inc.

ISO (2005), ISO/IEC 19503:2005 Information technology - XML Metadata Interchange (XMI).

Kayis, B., Arndt, G., Zhou, M., Savci, S., Khoo, Y., and Rispler, A. (2006), Risk quantification for new product design and development in a concurrent engineering environment. *CIRP Annals - Manufacturing Technology*, 55(1), 147-150.

Kayis, B., Zhou, M., Savci, S., Khoo, Y., Ahmed, A., Kusumo, R., and Rispler, A, (2007). IRMAS – development of a risk management tool for collaborative multi-site, multi-partner new product development projects. *Journal of Manufacturing Technology Management*, 18(4), 387 - 414.

Lutters, D., Mentink, R. J., van Houten, F., and Kals, H. J. J. (2001), Workflow management based on information management. *CIRP Annals - Manufacturing Technology*, 50(1), 309-312.

Ong, S. K., Xu, Q. L., and Nee, A. Y. C. (2006), Design reuse methodology for product family design. *CIRP Annals - Manufacturing Technology*, 55(1), 161-164.

Paige, R. F., Kolovos, D. S., and Polack, F. A. C. (2005), Refinement via consistency checking in MDA. *Electronic Notes in Theoretical Computer Science*, 137(2), 151-161.

Press, S. J. (1989), Bayesian statistics: Principles, models, and applications. John Wiley & Sons.

Rowe, G. and Wright, G. (1999), The Delphi technique as a forecasting tool: Issues and analysis. *International Journal of Forecasting*, 15(4), 353-375.

Rozenfeld, H. and Eversheim, W. (2002), An architecture for shared management of explicit knowledge applied to product development processes. *CIRP Annals - Manufacturing Technology*, 51(1), 413-416.

Saaty, T. (1999), Decision making for leaders: The Analytic Hierarchy Process for decisions in a complex world, New Edition 2001 (3rd ed.). RWS Publications.

Smith, P. G. (1998), Concurrent engineering teams. In D. Cleland (Ed.), Field guide to project management. USA: John Wiley & Sons.

Sperberg-McQueen, C. and Huitfeldt, C. (2000), GODDAG: A data structure for overlapping hierarchies. *Proceedings of DDEP-PODDP 2000*, 139-160.

Standards Australia and Standards New Zealand (2004), AS/NZS 4360:2004 - Risk Management.

Thannhuber, M., Tseng, M. M., and Bullinger, H. J. (2001), An autopoietic approach for building knowledge management systems in manufacturing enterprises. *CIRP Annals - Manufacturing Technology*, 50(1), 313-318.

Webster, D. (1988), Mapping the design information representation terrain. *Computer*, 21(12), 8-23.

Williams, M. E., Consolazio, G. R., and Hoit, M. I. (2005), Data storage and extraction in engineering software using XML. *Advances in Engineering Software*, 36(11-12), 709-719.

Winner, R. I., Pennell, J. P., Bertrand, H. E., and Slusarczuk, M. M. (1988), The role of concurrent engineering in weapons system acquisition. Alexandria, VA: Institute for Defense Analyses.

Zhou, M. (1996), An approach to product information prototyping. Ph.D., University of Leeds.