Extending the Model Driven Architecture with a pre-CIM level

Sheridan Jeary, Ali Fouad and Keith Phalp,

Software Systems Research Centre, Bournemouth University, United Kingdom

sjeary, afouad, kphalp @bournemouth.ac.uk

Abstract. Whilst the successful alignment of business strategy and IT development is an important topic, there are still few ways that this is possible. The Model Driven Architecture (MDA) shows promise as an approach but is focussed firmly in the IT domain at the level of the Platform Independent and Platform Specific Models. The Computation Independent Model (CIM) is targetted at business users, but this paper argues that the complexity of the CIM level disenfranchises them. The concept of a pre-CIM level could provide much richness to the MDA process and give the domain user greater ownership of the IT development that supports their processes.

Keywords: Model Driven Architecture; business strategy; CIM level;

1 Introduction

The importance of aligning Information Technology to business strategy and the difficulties inherent in the process were discussed by Henderson and Venkatraman in 1993 [1]. However, fifteen years later business is still in 'transition' in attempting to coordinate business process design and IT collaboration. Business managers are responsible for the design and re-design of business processes and expect the IT department to provide IT support for these re-designed processes. Conversely, IT departments are demanding that business managers understand the large and complex software systems which are supporting them in their every day tasks [2]. These issues are manifest in difficult projects where business managers struggle with employee concerns and IT managers have automation and integration concerns [2,3]. In addition to the business manager and the IT developer, there are a large number of stakeholders to any IT development and their views of both the prospective system and the system development process are very different [4].

The Model Driven Architecture (MDA) has been identified as having a useful application in the current business environment where competitive advantage is requiring business to respond rapidly to a fast changing environment. The business that can respond with agility and flexibility will reap the rewards [5]. The MDA provides a framework for the development and maintenance of software systems that allows an analyst to describe both business and software assets [6] but is heavily

weighted in favour of software assets. However, there has been little work in the area of model-driven development to encourage the business manager to invest time and effort into understanding the principles and practices. The MDA has a Computation Independent Model (CIM) level which is designed to enable the connection between the business analyst with a set of requirements and the IT architect with technical solutions [3]. At the CIM level, business process model is defined and is expected to capture the key business activities using a semi-formal language [7] such as ARIS-Event Process Chains, Business Process Modeling Notation (BPMN) or Vide CIM Level Language (VCLL) There are a substantial number of tools available that will allow a meaningful discussion to take place between various stakeholders and the developer so that full requirements can be obtained [8] but few of them are connected to or part of a specific instance of a model-driven architecture. There has however been some work into extending the MDA to include Requirements Engineering in respect of Software Product Lines [9] and Safety-critical systems [10].

The focus of this paper is on the gap between the business manager and the IT developer in the MDA. Our contention is that the CIM level semi-formal notation is too complex for a business manager to comprehend and is more useful to a business or IT analyst or a Requirements Engineer with a modeling background. Whilst this is perhaps of no consequence at an Enterprise level where a company employs a number of analysts to monitor their business and IT requirements; it is of consequence in smaller organizations (one could argue the majority of organizations) that have a number of business managers, an IT department of developers and no business analysts. We therefore believe that a pre-CIM level is necessary that allows the business manager to collect relevant informal information about the processes involved in their department and to allow them to begin to create informal models of those processes.

Section 2 gives a brief overview of the Model Driven Architecture whilst Section 3 describes the stakeholders involved in the MDA development process. Section 4 outlines the business process domain and the complexity of both process and models. Section 5 ties the disparate strands together and outlines some of our initial ideas about the format of the pre-CIM level. Following conclusions we outline on-going and further work we are carrying out in this area.

2   Model Driven Architecture

The Model Driven Architecture (MDA) is a conceptual framework, a set of standards and resource support provided by the Object Management Group to allow flexible system evolution, interoperability and interconnection [3]. It is based on the principle of model abstraction and transformation whereby a solution is described at an abstract level and using a series of automated steps is transformed into a concrete solution as an executable for a specific platform [5]. The MDA was developed from the principles of the traditional waterfall lifecycle which defines the process of software development into a number of phases including requirements, analysis,

design, coding, testing, deployment and maintenance. The amount of documentation and/or models between the phases particularly early in the life cycle add little value to the process unless they are constantly updated as later phases of the development cycle change the models. This is rarely done in practice. Similarly, the changes made to code during maintenance are rarely documented and what there is, is often of poor quality [11]. The Model Driven Architecture was thus developed in response to this, to create greater productivity by allowing the development of clearly defined model artifacts between the phases of the development lifecycle as can be seen in Fig. 1.
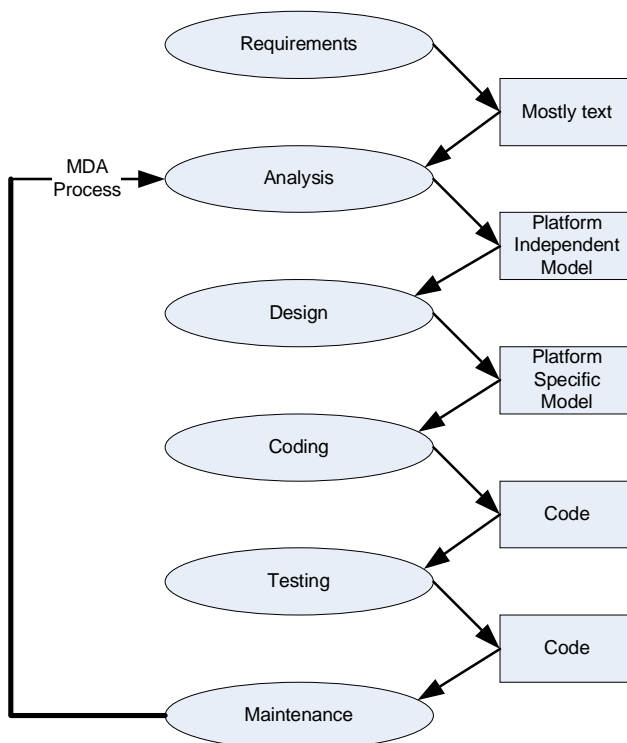
Figure 1:The MDA software development lifecycle (taken from [11])

The level of abstraction at which programmers work has risen over the years from the hardware and assembly code to high level language source code and the future is seen to be at the level of executable models [12]. The transformation allows the (mostly) automatic generation of model or code at the next level, thus giving major productivity gains. The Platform Independent Model (PIM) and the Platform Specific Model (PSM) are not seen as rigid levels or platforms with clear lines of demarcation so Fig. 1. could be seen as misleading in some respects as it implies that the PIM is always between analysis and design and the PSM is always between design and coding but as Brown points out "one person's PIM is another person's PSM"[3].

The emphasis in both research and the text books such as [11,12] is on the Platform Independent Model and the Platform Specific Model, there is little written about the Computation Independent Model (CIM). Indeed there are only two small sections in the MDA Guide, one of which articulates the role of the CIM:

"The CIM plays an important role in bridging the gap between those that are experts about the domain and its requirements on the one hand, and those that are experts of the design and construction of the artefacts that together satisfy the domain requirements, on the other" [13].

However, the Guide does not describe what form the CIM should take. Karow and Gehlert [7] agree and show that there is no model type in MDA covering the specification i.e. the source model for the system being built. Kleppe et.al. [11] discuss the CIM as a software independent model used to describe a business system. A number of PIM's may be specified from one CIM. They believe that automatic derivation of PIM's from the CIM is not possible because the choice of what part of the CIM to include is a human choice and therefore it is not possible to automate it. Therefore although it is part of the MDA, concrete information about what should or should not be in the CIM is rarely specified.

3   MDA stakeholders

Traditionally, the customer and user would be the only people identified as having a requirement on the system. However, it is a mistake to class users as a homogenous group. Two broader groups, containing a selection of roles are involved in any system development. Firstly, people on the development side, including: programmers, systems analysts, business analysts, project managers, senior IT management and the chief information officer. Secondly, there are those people from a business for whom the system is required [14]. Further definitions classify these users as individuals that utilise output or outcomes of an interaction with the system. These will include business users, business management and business strategy management. In addition, there may be external users, who are outside the boundary of the company, which the system will serve. For example, customers or potential customers, information users, trusted external users, shareholders or other sponsors (even the society at large), that are affected by the system.

Often, the stakeholders in the MDA process have been described solely in terms of software development team [15,16]. However, we argue in [4] that the CIM level roles are important, and that the domain user i.e. the person with the requirements on the system to be designed, is vitally important. The MDA Guide describes the CIM level role as being held by a domain practitioner who "*is not knowledgeable about the models or artefacts used to realize the functionality for which the requirements are articulated in the CIM*" [13].

The domain user is described by us in [4,17] as the end user of the constructed software solution. They work for the customer organisation and are an expert in their

special domain. Often they know about business economics and enterprise management but have normally only office application skills. Experience in modeling and business processes can not be assumed. For example, an insurance salesman knows about his company's offers and legal regulations and is supported by software solutions without any knowledge of their technical realisation. The domain user normally has no knowledge about business modeling but they are aware of what they need in terms of support for their daily tasks. In combination with a business consultant, the domain user will be able to articulate this process support requirement and various CIM level models can be constructed. The language and the graphical representation need to be easy to understand so that domain users can validate the correctness of the models. Since domain users usually use specific vocabulary, all tools should support translations using the user's vocabulary. The domain user also serves as a software tester for acceptance tests, i.e. reviews whether a simulated model performs the expected tasks.

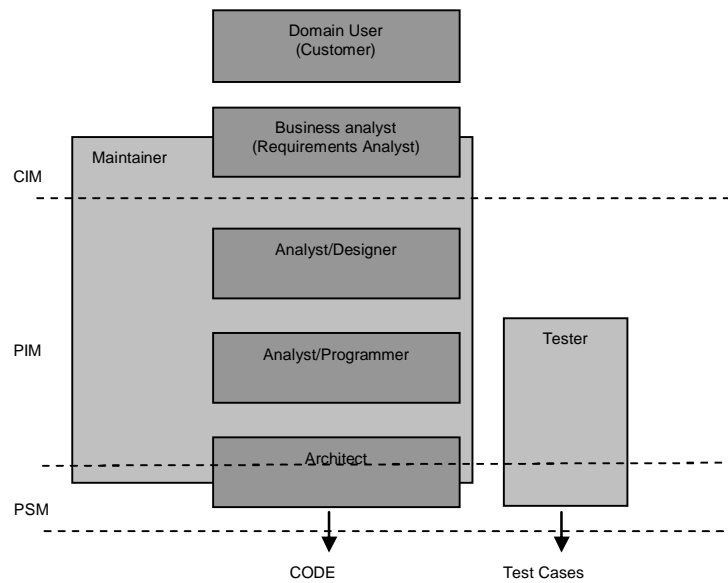The various roles and their correspondence to levels are shown in Fig. 2.

Figure 2: User roles and MDA levels taken from [17]

6   Sheridan Jeary, Ali Fouad and Keith Phalp,

## 4 The Business Process Domain

There have been a number of discussions as to what a business process is, and these are usefully summarized by [18]. For the purposes of this paper we use the definition given by Hammer and Champney which is selected both for its simplicity and because it uses vocabulary from the business domain [19]:

'*A business process is a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. A business process has a goal and is affected by events occurring in the external world or in other processes*'.

Business processes and their management have taken on new importance in recent years because successful process management allows enterprises to deliver value to the marketplace [20]. They deliver value by focusing the organization on the processes it performs and not the functional or operational divisions or departments that the organization consists of [21]. Increasingly business process management systems are the foundation upon which business applications are being built. The process, and the information that is stored about it, are the important factors; the process is visible and changeable. By focusing on the processes, making them flexible and changeable the organisation is now able to change the way it does its business often moving quickly to take advantage of today's business environment [22,23].

There have been a number of process modeling techniques proposed over time but only a few such as Event-Driven Process Chains [24] and Business Process Modeling Notation (BPMN) [25] have been widely accepted by the Business Process practitioner communities [26]. Many others with high expressiveness have remained only of interest to academia, such as advanced variants of Petri Nets [26]. The Object Management Group is taking a number of initiatives at this level, providing specifications for areas such as Business Rules and Workflow [27]. Process management systems must therefore have a capability that allows them to provide a view suitable to the specific role and needs of the user [22].

Whilst there are a number of different process modeling languages, the Business Process Modeling Notation (BPMN) [25] was designed to provide a common notation and potentially a standard. It was developed by revising many of the other process modeling languages and using their concepts as a guide, for example Unified Modeling Language (UML) Activity Diagrams [28], Integrated Definition Method (IDEF) [29] and Event-driven Process Chains [24]. BPMN is designed to be both understandable and usable to general business users, business managers, business analysts and technical developers. It also aims to be enough of a formal model to be easily translated into executable code [30].
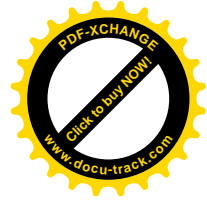
5   Why is a pre-CIM necessary?

Many processes that are part of a business organisation's value chain flow across multiple other organizations and the value is often in the end-to-end process thus enhancing the value chain across those multiple organisations. A good example adapted from [21] is the purchasing of raw material from suppliers and the use of them in the production of goods and services, the sales and delivery of these goods and services to intermediate customers and finally the sales and delivery of the goods to the final customers. In addition, there is the process of acquiring new customers from opportunities and the creation of new products according to customer needs. Monitoring and documenting the information inherent in that process is important. However, the process varies across the different organisations. There are process variants; private processes, customised processes for different partners and their suppliers, organisational best practices and standards. This all creates process complexity [22].

Because of this inherent complexity a business analyst needs a sound graphical language to capture that complexity [31]. Designing business process models is a difficult and often error prone task in that the models need to be easily and intuitively understood, but should not create ambiguity or allow incorrect inference to be made [32]. Barjis [20] considers both modeling and designing as complex tasks. Business process modeling languages provide users with an increased number of graphical constructs giving them greater expressiveness than earlier languages. But this is at the cost of language complexity [33]; the languages are cumbersome presenting the users with a large variety of constructs. Zur Muelen notes that Flowcharts in 1958 had 6 basic constructs and 4 extended constructs whereas BPMN in 2006 had 11 basic constructs and 39 extended constructs [33]. Therefore the CIM level of MDA has to allow for the complexity of both the real world and its processes and a complexity of modeling language which is in the domain of the business analyst.

There have been a number of discussions about the suitability of BPMN for Business Process Modeling. It is generally accepted that BPMN is not suitable for modelling organisational hierarchies and structures. It cannot map organisational resources, functional breakdowns, data and informational models, strategy or business rules [30]. Recker et.al. [34] conducted an ontological study using representational analysis and found a number of problems with BPMN. One of these findings was about the depiction of business rules that rely on state and transformation laws. A number of users questioned did not need to model business rules that rely on state because their Business Process Diagrams were intended for…. '*business users who would not understand such complicated diagrams*'… and …'*organizations required they use simpler diagrams to facilitate understanding*'. There was also confusion over how to model 'things' using such things as pools and lanes.

Whal and Sindre [30] analyse BPMN according to the Semiotic Quality Framework and have a number of comments to make. In particular, they believe that the goal of the notation being understood by non- technical business analysts and IT professionals is unrealistic. There are 23 different pre-defined elements to represent

different types of events. Most of the concepts have their origin in the IT domain and not the business domain. Wohed et. al. [35] examine the suitability of BPMN using a patterns evaluation framework. They find a number of ambiguities caused by the lack of formalization and similar issues with pools and lanes.

The Business Analyst at the CIM level is thus dealing with complex processes, often in a complex domain with a modelling language which is rich and graphical, but is created using the formalisms of the software domain and loses much of the richness of the business domain. Because of the use of formalisms that give the business process modelling notation its semi formal structure, it has become complex for business managers to understand and therefore a majority of them are not served by the CIM. In an environment where the domain knowledge is held by the business manager it would be sensible to capture it as part of the MDA process and thus enfranchise business managers to the MDA process.

We therefore feel that a pre-CIM level is necessary as part of the MDA. This level has none of the formalisms of the software domain or the semi formalisms of the business process domain, but will allow a business manager to capture information about his domain and use it to enable rich communication with both business analysts and systems analysts. In return it allows the analysts a vocabulary with the business manager and retention of the source of information from the domain. This will allow for increased traceability which will be useful in both the creation of new applications 'from scratch' and particularly when adapting and maintaining existing systems to support changing business requirements.

Therefore we find that we can adapt the earlier diagram shown in Figure 1 to include the pre-CIM and CIM. The pre-CIM holds informal business knowledge that will include organisational hierarchies, informal documentation, private process views, details of responsibilities, order forms etc. Where early identification of a requirement can be made, any relevant information can be identified and linked to allow early and very informal concept models to be made of relevant processes or sub-processes. These can be taken by the business analyst and a rich, business process model can be created at the CIM level using both their experience and the communication enabled by the pre-CIM level. This CIM will support a number of different PIM's.
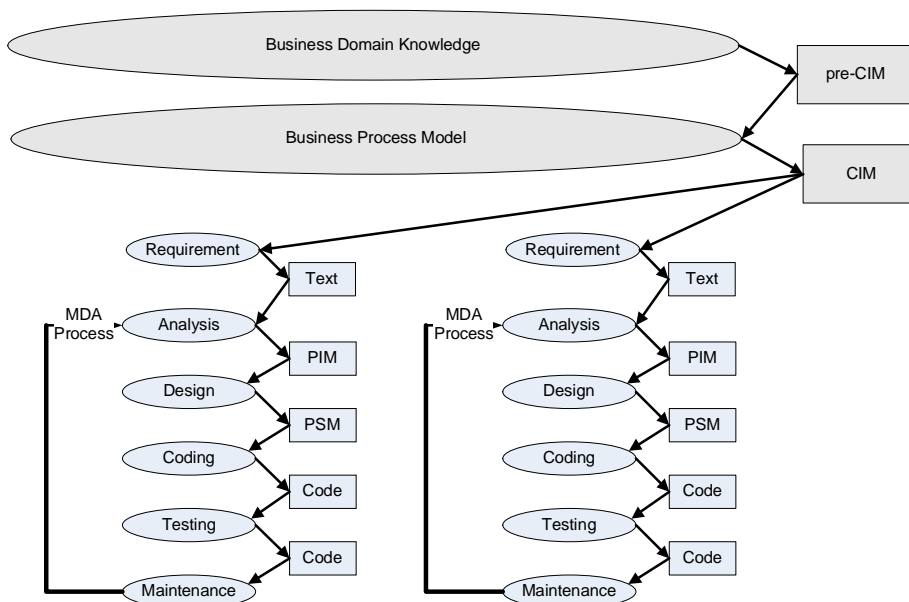
Figure 3: The MDA process including pre-CIM

Whilst we agree with Harmon [6] that the days when business analysts can create software from a process or sub process on a daily basis are not that close. We also agree that it is something that a business manager would not necessarily wish to attempt on a routine basis. However, we do believe that a business manager would have a large amount of domain knowledge that it makes sense to capture. In the short term there is still likely to be a gap between the models created by Business Process Modeling (BPM) toolsets and that created by MDA, But the BPM toolsets should provide some informal modeling techniques to allow the business manager to take ownership of his processes and the representation of his domain.

6   Conclusion

The gap between the business manager who is responsible for designing business processes and the IT developer who designs the IT systems that support those processes continues to be a challenge. It is particularly important when business is trying to align their business strategies to IT development and needs this alignment to allow them flexibility and agility to gain competitive advantage. Whilst it is possible that MDA may provide the means to allow this agility to happen, this work has shown that whilst the CIM contains enough information to inform the development of several PIM's most of the focus of the MDA is at the PIM level. The business domain is a world of complex processes and a rich and unstructured language that needs to be captured by the business analyst to take forward from the CIM to PIM levels. The

notation they use, whilst powerful, is not intuitive for business users. We believe that giving the domain user access to the MDA development process by creating a pre-CIM level will allow better communication, traceability through the whole process and mean that much of the richness of the business domain is captured.

7   Further work

The concept of pre-CIM is interesting and we are continuing work in this area. The importance of the business user to the future of software development should not be underestimated. We are working on a software tool as part of VIDE an EU 6th framework project which provides an end to end pre-CIM and CIM process as part of the MDA [36,37].

8   Acknowledgements

9   References

1.   Henderson, J. C. and Venkatraman, N. "Strategic alignment: Leveraging information technology for transforming organizations." in *IBM Systems Journal*, 1993, 32(1): 472-484.
2.   Harmon, P. *Business Process Change: A Managers Guide to Improving, Redesigning and Automating Processes* San Francisco, Morgan Kaufmann, 2003.
3.   Brown, A. W. "Model driven architecture : Principles and practice." in *Software Systems Modelling*, 2004, 3: 314-327.
4.   Phalp, K., Jeary, S., Vincent, J., Kanyaru, J. and Crowle, S. Supporting stakeholders in the MDA process. in 15th International Software Quality Management, 2007. Tampere, Finland.
5.   Brown, A. W., Iyengar, S. and Johnston, S. "A rational approach to model-driven development." in *IBM Systems Journal*, 2006, 45(3): 463-480.
6.   Harmon, P. "The OMG's Model Driven Architecture and BPM."   Retrieved 10 November 2004, from http://www.bptrends.com.
7.   Karow, M. and Gehlert, A. On the Transition from Computation Independent to Platform Independent Models. in Twelfth Americas Conference on Information Systems, 2006. Acapulco, Mexico.
8.   Wieringa, R. and Ebert, C. "RE' 03: Practical Requirements Engineering Solutions." in *IEEE Software*, 2004, 21(2): 16-18.

9.   Kabanda, S. and Adigun, M. Extending Model driven Architecture Benefits to Requirements Engineering. in SAICSIT 2006, 2006. Cape Winelands, South Africa.

10.   Biffl, S., Mordinyi, R. and Schatten, A. A Model-Driven Architecture Approach Using Explicit Stakeholder Quality Requirement Models for Building Dependable Information Systems. in Fifth International Workshop on Software Quality (WoSQ' 07), 2007. ICSE 2007, IEEE Computer Society.

11.   Kleppe, A., Warmer, J. and Bast, W. *MDA Explained: The Model Driven Architecture--Practice and Promise* Boston, Addison-Wesley Professional, 2003.

12.   Mellor, S. J., Kendall, S., Uhl, A. and Weise, D. *MDA Distilled: Principles of Model-Driven Architecture*. Addison-Wesley, 2004.

13.   OMG. "MDA Guide Version 1.0.1."   Retrieved 02 April 2003, from http://www.omg.org/mda/specs.htm.

14.   Avison, D. and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools*. London, UK, McGraw-Hill., 2006.

15.   Mellor, S. J. and Watson, A. "Roles in the MDA Process:MDA will make developers more productive not redundant."   Retrieved 26 April 2003, from www.omg.org/registration/registration-roles_mda.htm

16.   Aagedal, J. and Solheim, I. New Roles in Model-Driven Development. in Second European Workshop on Model Driven Architecture, 2004. Canterbury, UK.

17.   VIDE.*Standards, Technological and Research-Base for the VIDE Project, Project Evaluation Criteria and User Requirements Definition*. 2007, Framework 6 EU Commission IST033606STP

18.   Lindsay, A., Downs, D. and Lunn, K. "Business processes—attempts to find a definition." in *Information and Software Technology Journal*, 2003, 45: 1015-1019.

19.   Hammer, M. and Champney, J. *Re-engineering the Corporation; A Manifesto for Business Revolution*. New York, Harper Business, 1993.

20.   Barjis, J. "The importance of business process modeling in software systems design." in *Science of Computer Programming*, 2008, 71: 73-87.

21.   Giaglis, G. M., Paul, R. J. and Doukidis, G. I. Simulation for Intra- and Inter-organisational business modeling. in 1996 Winter Simulation Conference, 1996.

22.   Smith, H. and Fingar, P. *Business Process Management: The Third Wave*. New York, Meghan-Kiffer Press, 2003.

23.   Ould, M. A. *Business Process Management : A Rigorous Approach*. Swindon, British Computer Society, 2005.

24.   Keller, G., Nuttgens, M. N. and Scheer, A. W.*Semantische Processmodellierung auf der Grundlage Ereignisgesteuerter Processketten (EPK)*1992, Ver offentlichungen des Instituts fur Wirtschaftsinformatik, Heft 89 (in German), University of Saarland Saarbrucken

25.   OMG. "Business Process Modelling Notation Specification Version 1.1." Retrieved 2 May 2008, from http://www.omg.org/spec/BPMN/1.1/.
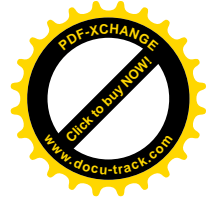
26. Recker, J. C. Why Do We Keep Using a Process Modelling Technique. in 18th Australasia Conference on Information Systems, 2007. Toowoomba.

27. OMG. "Catalog of OMG Business Strategy, Business Rules and Business Process Management Specifications." Retrieved 2 May 2008, from http://www.omg.org/technology/documents/br_pm_spec_catalog.htm.

28. OMG. "Unified Modelling Language Specification version 2.1.2." Retrieved 2 May 2007, from http://www.uml.org/.

29. NIST. "Integrated Definition for Function Modeling (IDEF0)." Retrieved 2 May 1993, from http://www.idef.com/IDEF0.html.

30. Wahl, T. and Sindre, G. An Analytical Evaluation of BPMN Using a Semiotic Quality Framework. in 10th International Workshop Exploring Modelliong Methods in Systems Analysis and Design (EMMSAD '05), 2005. Porto, Portugal.

31. Luttighuis, P. O., Lankhorst, M., van de Wetering, R., Bal, R. and van den Berg, H. "Visualising business processes." in *Computer Languages*, 2001, 27: 39-59.

32. van Dongen, B. F., van der Aalst, W. M. P. and Verbeek, H. M. W. Verification of EPCs:Using Reduction Rules and Petri Nets. in Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05), 2005. LNCS 3250, Springer-Verlag, Berlin.

33. zur Muehlen, M., Recker, J. C. and Indulska, M. Sometimes Less is More: Are process Modelling Languages Overly Complex. in 3rd International Workshop on Vocabularies, Ontologies and Rules for The Enterprise, 2007. Annapolis, Maryland., IEEE.

34. Recker, J. C., Indulska, M., Rosemann, M. and Green, P. How Good Is BPMN Really? Insights from Theory and Practice. in 14th European Conference on Information Systems, 2006. Goeteborg, Sweden.

35. Wohed, P., Van der Aalst, W., Dumas, M., Ter Hofstede, A. and Russell, N. On the Suitability of BPMN for Business Process Modelling in 4th International Conference on Business Process Management, 2006. Vienna, Austria., LNCS.

36. Kanyaru, J., Coles, M., Jeary, S. and Phalp, K. Using visualisation to elicit domain information as part of the Model Driven Architecture Approach. in First International Workshop on Business Support for MDA, 2008. Zurich, Switzerland.

37. Martin, A. and Loos, P. Software support for Compuational Independent Modelling in the MDA context. in 1st International Workshop on Business Support for MDA, 2008. Zurich, Switzerland.