Utilizing VDM Models in Process Management Tool Development: an Industrial Case

Claus Ballegård Nielsen

Aarhus School of Engineering, Denmark, clausbn@iha.dk

Abstract. Around the world companies are striving to become more effective and productive by improving their processes and workflows, within their organisation. Software tools exist, which aid and support process management and improvement methods. These tools are complex and difficult to build because they must ensure the integrity and consistency of the processes, while still being accessible, comprehensible and easy to use. This paper presents the work effort and the initial results of an academia-industry collaborative research project, which takes its foundation in the further development of an existing software suite, from a company developing advanced process management tools. An executable formal model has been created, via the Vienna Development Method (VDM), in order to analyse both the existing tool as well as an expansion, aimed at making the process management tools much more dynamic. The formal model was used to; (1) assist in the exploration of the system domain, (2) to aid the communication leading to more informed design decisions, and (3) to establish insight into complex dependency relations while still ensuring the consistency of the processes. The project involved a development scenario where the industry partner had no knowledge of formal modelling, and the academic partner had nearly no domain knowledge of the business field, therefore graphical representations have been linked to the model to aid the communication between stakeholders.

1 Introduction

In the ever-expanding commercial business market there is a constant push towards maximizing throughput and minimizing costs and development time in order to become more competitive, and preferably improve customer quality in the process. A key factor is increased efficiency, which can be accomplished by improving the processes and workflows within an organization. Software tools exist which are aimed at aiding and supporting process management and process improvement methods. The key focus of these tools is on establishing, maintaining and presenting process descriptions in an effective manner while still ensuring the integrity and consistency of the entire process.

This paper presents the work-effort and experiences gained through a research project which seeks to improve organization processes through the use of advanced tool support. The tool focuses on the knowledge that the end users of the process descriptions have and on the integration of concrete project data directly into the process.

These tools are complex and their inner behaviour is difficult to grasp, because they consist of many different data types which have a high number of relations in cyclic patterns, and there are imperative demands for versioning and traceability.

An existing tool, supplied by the industry partner Callis[1], is used as the baseline on to which the research effort is applied and a tool extension is to be developed. To clarify the complexities found in these tools and comprehend the effects on the tool that the intended extension will have, formal modelling has been utilized in the project. A formal model is used to analyse and describe the tool; as opposed to describing the business process descriptions themselves. The use of formal techniques for analyzing and validating software specifications and designs has been widely encouraged and extensively researched [5,15]. An executable model containing the key elements of the process management tool and the intended extension has been created, using the Vienna Development Method (VDM) [4,10,9]. By having an executable model the unclear parts and areas of concern can be analysed in a lightweight manner by modelling the functional behaviour, thereby adding more precision and confidence to the development process. VDM has been utilized in a wide variety of areas in the specification of software systems [3,13,14,2], however this is the first time VDM has been applied to a process management tool.

It has been anticipated that one of most important effects of having a formal model will be an increase in the quality of information exchange and communication between the project partners. This is especially important in the given project, because it consists of an industry partner with no knowledge of formal modelling, and an academic partner with nearly no domain knowledge of the business field.

Formal modelling has been used in numerous process management and modelling approaches [7]. Mishra et al [12] use the formal specification language CSP-CASL to integrate product and process quality. The applicability of formal specifications for realizing the objectives of process areas in CMMI is investigated, with the aim of contributing to the prospect of automation in process compliance. Van der Aalst [1] discusses the use of Petri nets for optimizing and supporting business processes in the context of workflow management. The graphical language and formally defined semantics of Petri nets ensures a clear and precise definition, that allows for the use of Petri nets as a design language for the specification of complex workflows. Despite the use of formal methods in these approaches in relation to processes, they are not directly related to the use of formal methods in this project. Most of the existing literature on the subject focuses on the formal method itself as a tool for describing and verifying the processes and not on specifying a software tool for supporting the process.

The remainder of this paper is set out as follows. Initially, Section 2 presents the notion of process improvement in the context of this paper. An outline of the research project is given in Section 3, while Section 4 supplies an overview of Callis' tool suite, in relation to which the work is performed. Section 5 contains the reasoning and expectations of using formal modelling as well as an outline of the performed work. The results and experience obtained are presented in Section 6, followed by concluding remarks in Section 7.

[1] Callis `http://www.callis.dk/`

2 Process Improvement

In the context of this paper a process is a well-established chain of procedures or subtasks which are to be performed to handle a certain task or operation. Companies and organizations use processes and process descriptions to aid the business in resolving tasks and fulfilling certain business objectives, through the means of a unified, structured and optimized workflow. Organizations are always searching for ways of improving their performance, and a well-established approach is to optimize the organizations existing processes, through process improvement.

In supporting the efforts in process improvement, key mechanisms include process methods, process standards and process management tools [6]. These tools aid organizations in defining and modelling their processes as well as publishing, broadening and maintaining existing business processes.

To the customers of Callis, the main responsibilities and goals required by these tools are:

Definition Aiding in the establishment and maintenance of the business process descriptions, as well as enabling interrelations between the organizations custom processes and standardized process descriptions such as ISO9001[2], Prince2[3] and CMMI[4].

Control Controlling the process description with regards to configuration, versioning and progress, in order to ensure traceability, evidence, integrity and metrics.

Communication Communicating the process to the users in an efficient, consistent and easy way, to promote and encourage the use of the process descriptions.

The establishment and management of a process generally involves multiple levels, which each have a role in relation to requirements and expectations of the process.

The overall process is defined and specified at a high level, which is detached from the individual projects which are actually the future users of the process, normally by management and process improvement agents. Subsequently the process description is passed down the chain in an organization and may be affected by process agents and managers at national, regional and divisional levels before finally reaching the project level. These levels entail that data in the process management tool may be affected at various levels, meaning that the tool must manage multiple versions, releases and users being related to data.

3 The Research Project

The research project is aimed at improving organizations business processes through the use of advanced tool support. The tool is to be improved by partly focusing on the knowledge and hands-on experience that the end-users of the processes descriptions

[2] ISO9001 `http://www.iso.org/iso/catalogue_detail?csnumber=21823`
[3] Prince2 `http://www.ogc.gov.uk/methods_prince_2.asp`
[4] CMMI `http://www.sei.cmu.edu/cmmi/`
(Accessed: May, 2011)

possesses, and partly by integrating concrete project data and artefacts directly into the process descriptions.

The *first* aspect of the project is to create an extension of the existing tool which supports and strengthens the end user's ability to interact with the tool, such that it can be used actively for establishing workflow overviews and concrete observations about process descriptions. Currently the process management tool is of a static natured, expect for a minor corner of the tool. Static means that the process descriptions are fixed and inactive, they appear as read-only descriptions of how a process should work. In one perspective this is essentially a desired behaviour which is required from a process manager and auditor standpoint. For reasons of traceability and appraisals the process description should be carved in stone at a specific version. However in another perspective this presents a challenge, given that the end-users who are actually using the process have no way of interacting with the tool. The knowledge of these end-users is extremely valuable to process improvement efforts because of their in-depth experience with the use of the processes, combined with their proficiencies and expertise in their field. This gives them the capability to identify both irregularities and ambiguities in a process description and to pinpoint elements of the process workflow that can be optimized.

Having inflexible and static processes creates a very rigid chain through which this information needs to pass before it gets worked into the projects process description. This has the consequence that feedback on the process description is lost, which eventually results in process improvement initiatives never occurring. Therefore the future is to expand the tool from being a tool for statically describing processes, into to a dynamic tool which allows for end-user involvement in improvements of processes and dynamic adjustments of a process. This will allow the end-user to become an active part in both the use and future-development of the process. The idea is to move the capabilities of making adjustments to a process, out to the projects and out to the people who are actually using the process. This is achieved by adding *tailorings* to the process description. A tailoring is the procedure of making minor changes, which has a limited project scope, to existing process descriptions through simple text descriptions. Through tailorings adjustments and comments to the process can then be looped back in to the general process or be used as optional examples or best practices *tailoring set* (a managed collection of tailorings) that can be added to other projects.

The *second* aspect of the tool's future is to supply an overview of the project progress based on the defined process and the artefacts specified in the process, with relation to a specific project. By probing the data in the actual files related to the artefact, the status can be read back into the tool and displayed in relation to the process in order to supply an overview of a projects progress. To enable the probing of project artefacts the tool needs to interface with various types of versioning document handling systems and document types. The goal is to create a full loop leading from process modelling and definition, through process usage, to process feedback, and back to including the feedback in the process modelling to complete the loop.

4 The Industrial Case - A Process Management Tool

The process management and modelling tool which is used as case in the project, is an industrial solution which is used by CMMI5 certified companies. The tool has support for the modelling and management of the process architecture, process integrity, and statistics.

The process descriptions are constructed by using a predefined set of standard description elements, into which organization specific data can be entered. The processes are defined by creating a structure of relationships between the description elements. In order to ensure the integrity of the process and to conform to process standards there are a large number of restrictions and rules as to how the description elements can be interrelated. There are a large number of different element and subelement types, which each have different properties; some types can reference themselves, some cannot, some types can reference other types but cannot be referenced by these types, some can only relate to certain types and some can only be referenced by certain types, and some types can only be used for grouping other types. This is illustrated in Fig. 1, which shows an abstract view of the elements in the tool. On the figure some of these relationships can be seen, for example DescriptionElementA is capable of referencing itself, DescriptionElementB can reference all subelement types and DescriptionElementC can reference all other types of DescriptionElements but only reference one type of subelement. To complicate things further each element can have multiple versions and processes can have multiple instances. This high number of relations, cyclic references, requirements and restrictions lead to a complex tool which is difficult to comprehend and get an overview of.

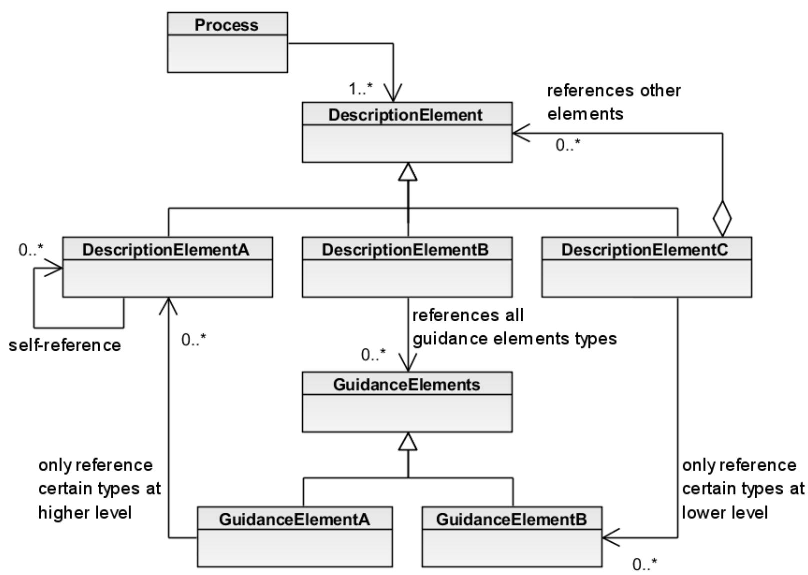

Fig. 1. Abstract class diagram of relations between elements.

5 Use of Formal Modelling in the Project

In order to manage and analyse the complexities of the existing tool and explore the possible solutions in extending the tool, in relation to the research aim, formal modelling has been used from the start of the project. The focus of the modelling has been on abstraction, understanding and communication and not on code verification. The model was developed by one person over a period of two months, with a weekly meeting between the project partners. The final model consisted of approximately 2500 lines of code distributed across 16 classes.

The creation of the model enabled the construction of different scenarios, through which the structures and relations of the elements in a process could be defined. This part of the model reflects the functionality of the existing tool and had three purposes:

- to gain insight into the existing tool and to better comprehend the methodology used in business process descriptions,
- to check the consistency of the existing tool and attempt to reveal unidentified problems,
- and to enable the modelling of the intended extension of the tool, in order to analyse the effects of incorporating the research goal in the existing tool implementation.

5.1 Exploration of the Domain and the Existing Tool

An executable model for a subset of the existing tool has been created in VDM++, with a focus on the structures used in the tool and the relationship between the process elements. Prior to starting the modeling, a smaller process was described using the tool, with the purpose of getting acquainted with both the methodology of business processes and to the concept and features of the tool. The existing source code of the tool was not used during the modeling, however the use of the tool and the central principles of process descriptions were discussed during weekly meetings with a process expert from Callis.

The ways in which the different element types can be combined, as mentioned in Section 4, are strictly controlled by the tool. As a process is built, it is type checked against an internal model of the allowed relationships between element types. Consequently the large matrix of constraints on references which exists between the different types was not included in the modelled subset. Instead the focus was on certain potential issues such as the risk of non-terminating recursion and on the structures which were relevant for the extension.

The classes were structured in a hierarchy which reflects the structure of the tool, but with a more abstract representation of the element types, as depicted in the class diagram in Fig. 2. This class hierarchy enabled the modelling of key elements of both the existing and future tool, and fairly elaborate processes could be defined through the use of the model. In the class diagram it can be seen that the element type is capable of referencing itself and other elements of the same type, thus enabling recursion.

The *ProcessDecorator* and *GuidanceDecorator* classes are part of the Decorator pattern [8] and are used to model future aspects of the tool's future. The *GuidanceDecorator* is used to add tailorings to elements at the lowest level of the tool hierarchy. The

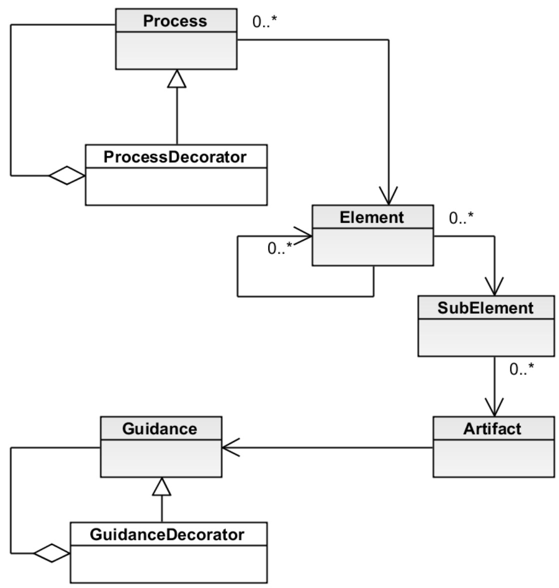

Fig. 2. Class diagram of the models class hierarchy for describing processes

use of the decorator allows for tailoring to be added multiple times to the same element, without directly changing the existing element; this is in accordance with the intended functionality of the future tool.

With regards to tailoring of the process, a tailoring is actually detached from process versions in the future tool. Here the *ProcessDecorator* can be used for altering a process at the top most level, by removing existing elements or by adding new. This was used for creating new versions of an existing process, in order to examine how tailoring were to be handled between different versions the process; for instance what happens to a tailoring that has a relation to a process element, which is then removed in a newer version of the process.

The model allows for different scenarios to be defined and run in order to study diverse process constructs, both in relation to complexity and size. Instances of the class structure in Fig. 2 can be created in a scenario, such that different object structures can be characterized. The capability of running different scenarios is implemented through the use of the Strategy pattern [8], were each concrete strategy represents a scenario. The use of the strategy pattern allowed the scenario to be selected at runtime by changing a configuration file, thus allowing for different scenarios to run without having to change the model itself.

To get an overview of the structures for at given scenario, the Visitor Pattern [8] was used to traverse the object hierarchy and print the object data. Different implementations of the Visitor class were used to focus on certain aspects of model, for instance one was used to present structures relating to tailoring and one was used for constructing a graphical representation of the running scenario. The graphical representation is described in Section 5.2.

An excerpt of a scenario output is shown in listing 1.1, where the structure of the scenario can be seen by each element name being printed at a certain depth. The digit following the # denotes the elements unique ID.

```
###-- Displaying process overview

-Process Set-
----#3 The Super Process 1
--------Elements
----------#4 Element 1
------------SubElements
--------------#5 SubElement 1.1
--------------#8 SubElement 1.2
--------------#10 SubElement 1.3
----------#11 Element 2
------------SubElements
--------------#12 SubElement 2.1
```

Listing 1.1. Excerpt of Scenario Output

By applying a special Visitor, made specifically for tailoring, an output can be created which displays the path to tailored elements, as depicted in Listing 1.2. The name in the parenthesis indicates the source of the tailoring. This output displays a new version of the process set previously shown in Listing 1.1, in which certain elements have been tailored.

```
###-- Displaying tailored elements

-Process Set-
----#18 The Super Process 1 version 2
--------Elements
----------#4 Element 1
------------SubElements
--------------#5 SubElement 1.1
                Tailoring test text (HealthCare Project)
--------------#8 SubElement 1.2
                Check X before that (The SuperProject)
```

Listing 1.2. Excerpt of Scenario Output from Tailoring Visitor

5.2 The Model as a Means of Communication

Having the VDM model enabled different scenarios of distinctive process setups to be defined and analysed by executing the model. However during the development process we found that it was still difficult to discuss the details of the model and to get a quick overview of the effects that different development suggestions may have on the tool. This was especially true for the industrial partner with no prior knowledge of formal modelling or VDM. Initially some difficulties were caused by the difference in expertise with regards to formal modelling and process descriptions between the industrial and the academic partner respectively. The limited cross-knowledge made it challenging to determine what the modelling could and should be used for. Several times areas of the tool extension, which were considered candidates for analysis in the model, were eventually discarded as they were revealed to mainly concern how data should be selected and managed from a user-interface and usability viewpoint. A subject for which formal modelling is of limited use. This may have been caused by the industry partner's limited understanding of the capabilities of VDM modelling. Likewise the academic partner had a challenge in understanding the process descriptions, because of the narrow knowledge of the business domain. Since the tool is built to support multiple business areas and be compliant with multiple standards it was difficult to get an overview of the constraints in the process descriptions, especially because no formal definition of the processes exists, except partially what can be derived from the entire ISO9001 specification or CMMI models. Instead the construction of the tool is based on de facto standards and experience in how the process description elements are used in practice by process agents.

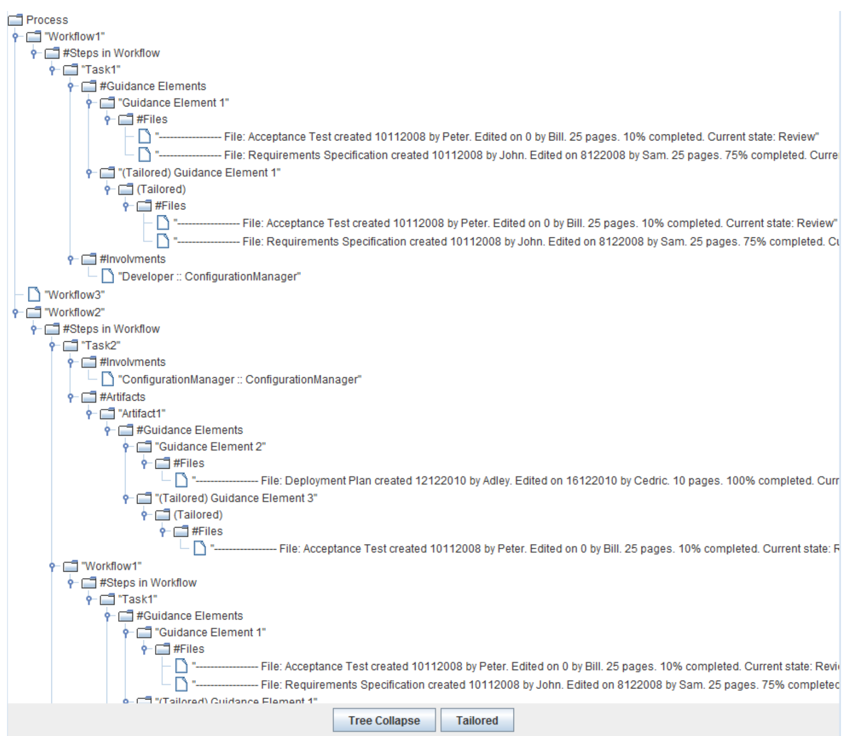

Fig. 3. Screenshot of the graphical presentation of the model

To aid the communication and momentum in the weekly development meetings, a graphical representation was made for a portion of the process structure. This is possible owing to a functionality in Overture [11] which enables the developer to utilize functionality defined in an external Java jar file. It is possible to add a graphical representation on top of the model, which can then be completely controlled from within the VDM model. An example of such a representation is depicted in Fig. 3, where the structure of a process is displayed in a Java JTree structure. Buttons at bottom of the screen enable a switch from a view which focuses on the entire process to a view that focuses on tailored process elements.

6 Results

A key goal of having the model was to use it in relation to the two research aspects and the extension of the tool. Tailoring was designed to strengthen the end-users possibility of interacting with the process, and the model made it possible to (1) examine how different sources of tailoring could be added to a single process and (2) to determine the behaviour of tailoring when applied to different versions of the same process. With regards to the aspect of relating actual project specific data to a process, the model was used for identifying the required data and to examine if the tool would be capable of retrieving the required data. This lead to some further investigation, outside the scope of the modeling, as to how this specific data could be gathered and described such that the tool could process it.

The model did not disclose as many issues as were initially expected, neither in the existing tool nor in the scenarios run for the extension. The only real defect discovered in the existing tool was a risk of recursion in certain scenarios. This is however an issue which is fairly simple to predict, considering the design of the tool. The existing tool has a check for self-referential recursion, but the recursion issue found, was not considered in the existing tool. Nonetheless as it turned out, it was not a risk either as the tool has no automatic traversal of references, meaning the user of the tool would have to traverse the recursive path manually to encounter the problem, and additionally the tool automatically limits its depth to a certain level.

In relation to the extension, the model did not reveal any faults or inconsistencies that could be a cause of concern in a later implementation of the tool. Instead the model confirmed the behaviours and effects that had been anticipated during the continuous planning of the extension. In this sense the model provided assurance in the development process, as it made it possible to test different scenarios, which could then be examined and the expectations could be confirmed.

There were parts of the tool that it was not necessary to include into the model, because they were clarified sufficiently. This has to be attributed to the weekly meetings with the domain experts from Callis, which provided precise and valuable insight into the tool and process modelling in general. Without this knowledge the model would have been substantially larger, because of the many additional unknowns that had to be included and examined. Thus good communication is extremely important and with the project taken as a whole, the model was mainly used as mean for improving communication and driving development forward. The addition of a graphical representation of modelled scenarios improved and simplified communication between the project members, because it made the modelled scenarios independent from the formal notation used. The visual examples made it easier for Callis to provide feedback on questions raised from the concrete work with the model, leading to a minimization in the risk of requirement misinterpretations.

There were however limits to the areas in which the model was applicable. A lot of the subjects that the industry partner had an interest in modeling, had to do with human-computer interaction and data presentation, e.g. how does one define tailoring for an element or how is data from an artefact presented. These are questions and topics, to which the VDM model could not be used, and more traditional development techniques had to be brought in to play.

7 Concluding Remarks

Specifying and developing new functionality for an existing tool is a challenging and difficult task. Not only must the challenges faced in the development of completely new tools be handled, but the impact on, and the influence of, the existing tool has to be included as well. Anticipating how new additions will affect existing software requires knowledge and insight in to both the business area and the software tool itself. This paper has presented the experiences gained in the use of VDM modelling in the development of a process management tool supplied by an industrial partner. A model has been created which includes a subset of the existing tool, as well as elements of an extension, planed for a future versions of the tool.

The use of the model did not disclose as many issues as were initially expected, and with the exception of a non-terminating recursion risk related to circular references, only minor issues were found. The creation of the model did however provide an insight into the tool and the business domain, which was advantageous given the academic partners limited domain knowledge of the business field. Similarly a feature of VDM was used to create a graphical representation of the model, which made it possible to interact with the model without having any previous knowledge of formal modeling. This made it easier to communicate the modelled scenario between the project partners. In this project the principal value of the model has been the increased communication in the project. The success of a project is very dependent on the communication between the domain expert and the developers. Here the VDM model was successfully used as a means of discussing specification of functionality and in achieving consensus in order to avoid misunderstandings and ambiguities.

The future plans involve a full implementation of the extension into the existing tool, and the deployment of the tool in industry workshops, in order to evaluate the effects on process improvements.

Acknowledgements The author would like to thank the industrial partner Callis for taking part in this research project and in this publication on the project experiences and findings. In particular the development meetings and discussions with Peter Voldby

Petersen have been highly valuable in the comprehension of the Process Management Tools domain and the end-users application of these types of tools. This research project is partially funded by the iKraft Project, as part of Innovation pool in the Central Denmark Region.

References

1. Aalst, W.M.P.V.D.: The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers 8(1), 21–66 (1998)
2. Agerholm, S., Schafer, W.: Analyzing SAFER using UML and VDM++. In: Fitzgerald, J., Larsen, P.G. (eds.) VDM in Practice. pp. 139–141 (September 1999)
3. van den Berg, M., Verhoef, M., Wigmans, M.: Formal Specification of an Auctioning System Using VDM++ and UML – an Industrial Usage Report. In: Fitzgerald, J., Larsen, P.G. (eds.) VDM in Practice. pp. 85–93 (September 1999)
4. Bjørner, D.: The Vienna Development Method: Software Abstraction and Program Synthesis, Lecture Notes in Computer Science, vol. 75: Math. Studies of Information Processing. Springer-Verlag (1979)
5. Clarke, E.M., Wing, J.M.: Formal Methods: State of the Art and Future Directions. ACM Computing Surveys 28(4), 626–643 (1996)
6. Conradi, R., Fuggetta, A.: Improving Software Process Improvement. IEEE Software 19(4), 92–99 (July 2002)
7. Curtis, B., I.Kellner, M., Over, J.: Process modeling. Commun. ACM 35(9), 75–90 (September 1992)
8. E.Gamma, R.Helm, R., J.Vlissides: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series, Addison-Wesley Publishing Company (1995)
9. Fitzgerald, J.S., Larsen, P.G., Verhoef, M.: Vienna Development Method. Wiley Encyclopedia of Computer Science and Engineering (2008), edited by Benjamin Wah, John Wiley & Sons, Inc
10. Jones, C.B.: Systematic Software Development Using VDM. Prentice-Hall International, Englewood Cliffs, New Jersey, second edn. (1990), iSBN 0-13-880733-7
11. Larsen, P.G., Lausdahl, K., Ribeiro, A., Wolff, S., Battle, N.: Overture VDM-10 Tool Support: User Guide. Tech. Rep. TR-2010-02, The Overture Initiative, www.overturetool.org (May 2010)
12. Mishra, S., Schlingloff, B.H.: Compliance of cmmi process area with specification based development. In: ACIS International Conference on Software Engineering Research, Management and Applications. pp. 77–84. IEEE Computer Society, Los Alamitos, CA, USA (2008)
13. Schlatte, R., Aichernig, B.: Database Development of a Work-Flow Planning and Tracking System Using VDM-SL. In: Fitzgerald, J., Larsen, P.G. (eds.) VDM in Practice. pp. 109–125 (September 1999)
14. Smith, P.R., Larsen, P.G.: Applications of VDM in Banknote Processing. In: Fitzgerald, J.S., Larsen, P.G. (eds.) VDM in Practice: Proc. First VDM Workshop 1999 (September 1999), available at www.vdmportal.org
15. Woodcock, J., Larsen, P.G., Bicarregui, J., Fitzgerald, J.: Formal Methods: Practice and Experience. ACM Computing Surveys 41(4), 1–36 (October 2009)