Results of a case study on process-oriented knowledge management

Mirjam Minor, Daniel Schmalen
University of Trier
D-54286, Trier, Germany
{minor,schmalen}@uni-trier.de

Andreas Koldehoff
Silicon Image GmbH
Garbsener Landstr. 10
30419 Hannover, Germany
andreas.koldehoff@siliconimage.com

Abstract

In this paper, we present a case study on applying our novel, agile workflow technology in a process-oriented knowledge management approach to Silicon Image's real chip design project MARVIN. The subjects of interest are whether the technology is powerful and flexible enough to represent the design flow of MARVIN, whether the prototypical implementation can handle MARVIN's change requests, and how the chip design experts assess the benefits and potentials of applying such a technology to their area of expertise.

1 Introduction

According to Moore's law, the number of transistors on a chip should double in each technology generation [Jansen, 2001, p. 1]. For more than 30 years, the market development is in full compliance with this law. The rapidly growing number of components (design units) that can be integrated on a single chip has a strong impact on the design process, which is getting increasingly complex. At the same time, chip design processes are on a tight time-to-market schedule. Each delay of delivery causes raised costs or even the danger to forfeit a market segment. A large number of people and tools is required in order to remain on schedule. Under the constraints of process complexity and time pressure, it is a very difficult task to handle change requests, i.e. to deal with modifications of the customer requirements or of the internal goals and conditions of a chip design process. Conventional change request documents alone are not any more a good instrument for not loosing control over the processes. A process-oriented knowledge management approach that employs novel, agile workflow technology [Minor, 2008] promises relief. It supports the team members of chip design projects by enacting a structured workflow based on a step to step description of the design process called *design flow*. In contrast to a conventional workflow, the agile workflow remains flexible when being enacted so that it can be adapted structurally whenever change requests occur.

In this paper, we present the results of a case study that we conducted on applying agile workflow technology at Silicon Image GmbH in spring 2008. The paper is organized as follows: In Section 2, we briefly introduce the novel, agile workflow technology based on our previous work. In Section 3, we describe the setup of the case study including a description of the regarded chip design project MARVIN. Furthermore, the results of the case study are given. Finally, Section 4 discusses the results.

2 Agile workflow technology

Workflows are 'the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules' [WFMC, 1999, p. 8]. A *workflow management system* 'defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications' [WFMC, 1999, p. 9]. Agile workflow technology [Weber, 2005; Minor, 2008] is a novel workflow technology that allows the adaptation of ongoing workflows. Our approach focusses on ad-hoc changes and late modelling of already enacted *workflow instances* that have been derived from process templates called *workflow definitions*.

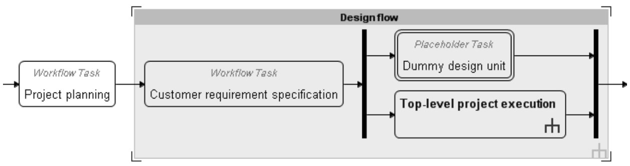

Figure 1: Clipping of a workflow definition for a chip design flow.

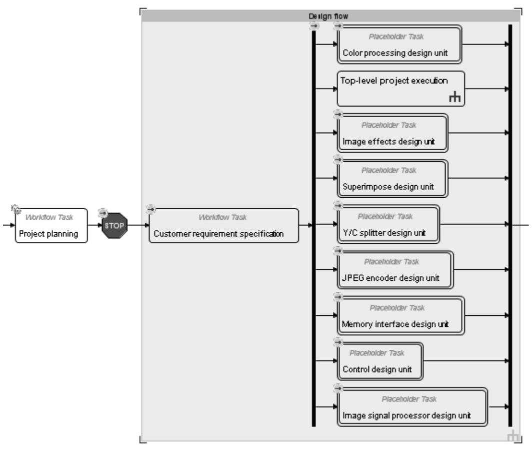

Figure 2: Clipping of the ongoing MARVIN design flow.

Figure 1 depicts a clipping of our prototypical workflow modelling tool with a part of a sample workflow definition that specifies the design flow. It is in UML activity diagram notation [Stoerrle, 2005]. The sequential activities 'Project planning' and 'Customer requirement specification' are followed by the activities for designing the building blocks of a chip (*design units*) in parallel to those for integrating the design units on a single chip ('Top-level project execution'). The 'Dummy design unit' is a placeholder for a sub-workflow that is supposed to be replaced later on. Figure 2 shows the real design units of the sample chip design project MARVIN (see Section 3.1). MARVIN's design units have been specified during the 'Project planning' activity. This replacement of the 'Dummy design unit' is a sample for late modelling. Typical ad-hoc changes are, for instance to re-order some parts of a workflow instance or to insert an additional activity.

2.1 Workflow modelling language

We have specified a control-flow-oriented workflow modeling language for agile workflows. The language has the five basic control flow elements sequence, AND-split, AND-join, XOR-split, and XOR-join, and also loops. We regard loops as structured cycles with one entry point to the loop (the control flow element LOOP-join) and one exit point from the loop (the control flow element LOOP-split).

For adaptability reasons, we have created two more control flow elements: breakpoints and placeholder for sub-workflows. Breakpoints are necessary for the control of modifications in a workflow instance concurrently to the execution. Setting a breakpoint prevents the workflow engine from overrunning activities that are about to be modified. Placeholder for sub-workflows stand for a reference to another workflow instance that is enacted when the control flow reaches the placeholder. For further details on the modeling language, we refer to the literature [Minor, 2008].

2.2 Workflow management system

The workflow modeling language facilitates the agility within workflows. This leads to new requirements for the workflow execution that can not be met by traditional workflow enactmentservices. The two additional control flow elements introduced for breakpoints and for sub-workflows need to be handled. Furthermore, the loop blocks require a special treatment since an adaptation of an ongoing loop may lead to different iterations of the same loop.

We implemented a prototype of an agile workflow management system (WFMS) supporting the above requirements. Its architecture is depicted in Figure 3. The WFMS consists of three parts: the user interfaces, the workflow enactment service with the underlying data access layer, and the test engine.

The user interfaces are the following:

- The modelling tool provides a graphical user interface to create and adapt agile workflows.
- The work list shows the activities that have been assigned to a particular user and notifies the workflow enactment service when an activity has been finished.
- The admin tool is for administrative purposes like restarting the workflow enactment service.

The workflow enactment service consists of the following sub-components:

- The communication broker uses Web technology for bi-directional message transfer.

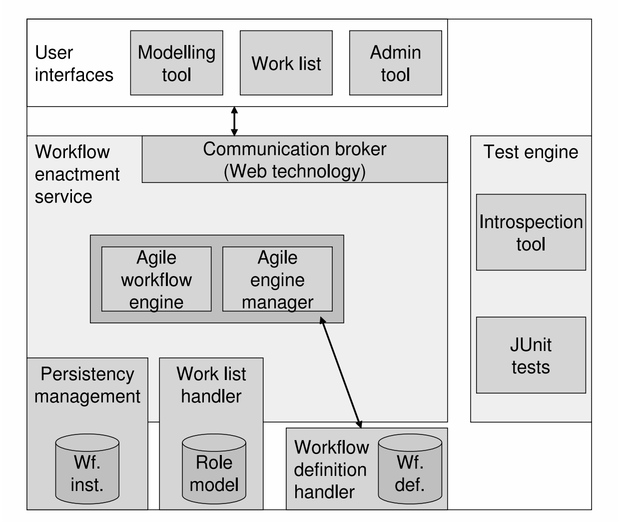

Figure 3: Architecture of the WFMS.

- The core of the workflow enactment service consists of an agile workflow engine and an engine manager. They control the execution and adaptation of the agile workflows.
- The persistency management cares for the consistent and persistent storage of the workflow instances at run-time.
- The work list handler manages a role model.
- The workflow definition handler provides workflow templates.

The test engine for the left hand side of the architecture contains the following sub-components:

- The introspection tool monitors the internal execution and control flow of the agile workflow engine. It allows also to modify a workflow instance for debugging purposes.
- The JUnit tests are for further testing activities.

The prototypical implementation has been used for conducting the case study. Additionally, it has been one of the subjects of interest during the case study.

3 Setup and results of the case study

Proof of concept is the primary goal of the case study. This includes a first assessment of the benefits and potential of applying agile workflow technology to chip design.

The research questions cornerning the novel, agile workflow technology have been the following:

- Will the workflow modelling language be powerful and flexible enough to describe the design flow of a real chip design project?
- To what extent will the prototypical implementation of the agile WFMS be able to deal with the process adaptations due to change requests?
- How will the chip designers appraise the benefits and future potential of agile workflow technology?

We investigated the first two topics by means of the real chip design project MARVIN that will be described in the following. The third question was answered by the chip experts via a subsequent questionnaire.

3.1 MARVIN

The Megapixel Camera Interface DesignObject^{TM} (MARVIN) is a camera interface supporting up to 12.6 megapixel video and still picture input data. It is mass-produced, for instance for mobile phones with integrated cameras. MARVIN can easily be adapted to lower or higher image resolution, for instance for saving gate count or memory. Different MARVIN implementations (MARVIN-3MP, MARVIN-5MP, MARVIN-12MP) are available. For our experiments, we have chosen a MARVIN adaptation project with a duration of twelve months. Five developers worked part-time at this project with an overall effort of about three man-months.

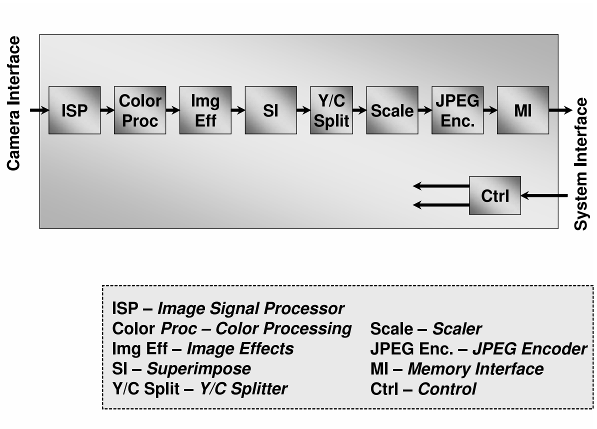

Figure 4: Block diagram of the MARVIN camera interface.

Figure 4 depicts a block diagram with MARVIN's design units. The chain contains image processing, scaling and compression functions: The Image Signal Processing (ISP) block samples the image received via the camera input. The Color Processing module (Color Proc) is designed for color adjustments. A set of image effects (Img Eff) is supported like sepia, grayscale, negative, or sketch. The superimpose (SI) module overlays an image with a bitmap from the main memory in order to give resize support. The Y/C Splitter module (Y/C split) separates the pixel data into their luminance and chrominance components. The scaler (Scale) down-scales the image to the resolution needed for capturing, viewfinding or encoding. The scaler uses separate scaling engines for luminance and chrominance processing. The hardware JPEG encoder (JPEG Enc.) produces a JPEG data stream. The control unit (Ctrl) allows the host CPU access to a set of configuration registers. The memory interface (MI) is responsible for writing the image data stream color component separated into system memory.

3.2 Change requests

We investigated the initial design flow and the seven change requests that occurred during the MARVIN adaptation project. A sample change request was that during running the validation software it turned out that a lookup table has been designed too small. The real test data from the customer required a bigger table than he had specified before. The validation happens quite late during project execution namely when preliminary syntheses results (synthesis = transformation of the logical description into a chip layout including the placement of gates) of all design units are available. So, integrating this change request with the nearly completed design process was a difficult task. The ongoing design steps refining the particular design units had to be coordinated with the augmentation of the lookup table. We studied for all change requests including this sample whether it could be handled by both the modelling language and the prototypical WFMS.

3.3 Results of the case study

The results of applying the agile workflow technology to MARVIN's change requests have been the following: The workflow modeling language is powerful enough to represent the initial MARVIN design flow and its multiple adaptation due to the change requests (compare the dark bars in Figure 5). The light grey bars show to what extent the prototypical WFMS is able to deal with the workflow adaptations. The above sample change request was solved by an additional parallel branch in the AND block depicted in Figure 2. This branch describes the activities in order to resize the lookup table. As these activities have not been blocking the ongoing refinement of design units, a synchronisation of the results was not necessary explicitly but was done in conjunction with the second run of the validation software. As Figure 5 shows, the implementation was not fully able to handle the changes in two cases: In change request 6, the implementation of the modelling GUI had problems to display the second iteration of a loop correctly so that it could not be adapted appropriately. However, the introspection tool was able to modify the second iteration, and the workflow enactment service executed it correctly. In change request 7, a modeling mistake in the template led to a missing loop. Instead of repeating a part of the workflow, it had to be copied. This turned out to be a laborious task as copy and paste is not yet supported by the modelling tool.

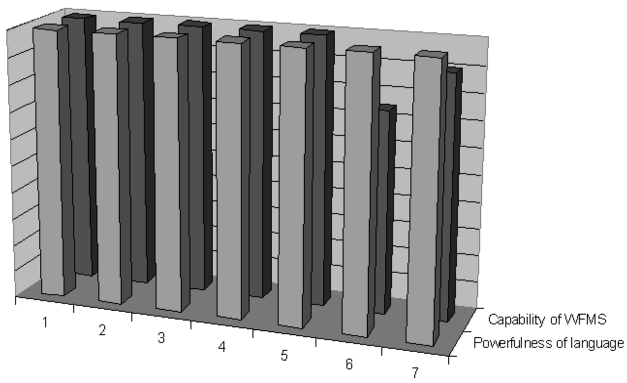

Figure 5: Powerfulness of language and capability of WFMS to handle change requests.

The expert answers to the questionnaire led to the following main insights:

a) There is a potential of the agile workflow technology to decrease the efforts for coordinating large teams.

b) There is a potential of the agile workflow technology to improve the inter-organisational co-operation in longtime collaborations.

c) The technology is not yet mature enough to handle all change requests without any problems.

ad a): The estimated time spent for coordination in chip design projects is 10 to 20 % of the overall efforts. A chip expert saw a potential for the agile technology to reduce these efforts especially for large teams and for the integration of new staff members.

ad b): The application of agile workflow technology for the collaboration with customers was assessed positively for longtime relationships only. For shorttime customers, the efforts were expected to beat the benefits.

ad c): The experts criticized the maturity of the prototype concerning unexpected loops. This was in accordance to the above observations with the missing copy and paste support for handling change request 7. Alternatively, we will think about a new feature to insert loops that go to the past. However, this would require major implementation efforts as our prototype does not allow changes of the 'past' except for inserting an additional branch into an active AND block.

4 Discussion and conclusion

The results of applying agile workflow technology to the chip design project MARVIN in a case study are satisfactory. The case study including the handling of real change requests and a subsequent questionnaire to the experts has been successful. It has shown that our concepts of novel, agile workflow technology are applicable to chip design processes and have promising potentials. The prototypical implementation is working in principle but should further mature, for instance requires an increased usability by a copy and past support. The chip experts identified the biggest, potential benefits of our approach in a) an increasing productivity by decreased coordination efforts and b) an improvement of the position in competition by better longtime customer relationships.

The case study is an important step towards a commercial development of an agile WFMS based on the above presented concepts, towards the transfer of our approach from the chip design area to further application areas, and towards the development of further methods and concepts of agile workflow technology. For first extensions concerning the support of change reuse we refer to the literature [Minor, 2008]. Our future vision includes an automatic adaptation of workflow instances based on experiences from handling previous change requests.

Acknowledgements

The authors acknowledge the Federal Ministry for Education and Science (BMBF) for funding the URANOS project under grant number 01M3075. We acknowledge the assistance we have received from the chip experts Marko Höpken, Frank Recoullé, and Stefan Pipereit, all Silicon Image.

References

[Jansen, 2001] Dirk Jansen. Einführung. In: D. Jansen (Ed.), *Handbuch der Electronic Design Automation*, Carl Hanser Verlag, München, 2001.

[Minor, 2008] Mirjam Minor and Alexander Tartakovski and Daniel Schmalen and Ralph Bergmann. Agile Workflow Technology and Case-Based Change Reuse for Long-Term Processes. *International Journal on Intelligent Information Technologies*, 4(1):80–98, 2008.

[Stoerrle, 2005] Harald Störrle. *UML 2 für Studenten*, Pearson Studium Verlag, München, 2005.

[WFMC, 1999] Workflow Management Coalition. Workflow management coalition glossary & terminology. *Retrieved from the Web May 23, 2007,* http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf.

[Weber, 2005] Barbara Weber and Werner Wild. Towards the Agile Management of Business Processes. In: K. Althoff, A. Dengel, R. Bergmann, M. Nick, T. Roth-Berghofer (Eds.), *Professional Knowledge Management, Third Biennial Conference, WM 2005, Kaiserslautern, Germany, Revised Selected Papers*, LNCS 3782, pp. 409 – 419, Springer, 2008.