MILOS and MASE: Past & Present

Compiled by
Michael M. Richter
with some additions by
Frank Maurer

1 Preliminary Remarks

The purpose of this article is to describe the Process-Centered Software Engineering Environment MILOS (*Modeling Language and Operational Support for Software Processes*; the interpretation of the acronym was later on changed *to Minimally Invasive Long Term Organizational Support for Software Development*).

We explain its main concepts and features, the major scientific results, the impacts on academic education, the relations to industry, the history of the MILOS project, the present state and the future prospects. There are quite many publications on MILOS but due the fact that they are written by different authors on different occasions they do neither have a unified view nor a unified terminology. Therefore we try to give a compact and consistent representation of what happened.

Although MILOS is grounded on two modeling approaches by the groups of Professor Richter and the group of Professor Rombach at the University of Kaiserslautern, we put emphasis on the first one because we have more insights here. It should be observed that MILOS is still a very active area of research.

If one records the evolution of ideas over a long time it is natural to see change of ideas, goals, concepts etc. These aspects will play a role in this summary.

In section 2 we will mainly document the development and list the main events without discussing the content. This will be done in section 3 where the concepts and methods are discussed – also in the order they have been developed historically.

2 Historic View: Persons, Events, Projects, Degrees

2.1 The Beginning and the Early History: CoMo-Kit

The roots of the MILOS-Project in one sense go back to 1992. At that time, the group of Professor Michael M. Richter at the Computer Science Department of the University of Kaiserslautern was working in the area of knowledge based systems. Due to the application oriented view, much work was investigated in application projects with partners. For this reason knowledge engineering played an increasing role.

Motivated by this, Frank Maurer started the CoMo-Kit-Project dealing with flexible knowledge engineering that essentially resulted in his dissertation in 1993. The subsequent thesis of Gerd Pews (1999) dealing with Urban Land-Use Planning relied heavily on the use of CoMo-Kit.

Projects:

Urban Land-Use Planning, founded by the VW- Foundation.

Theses:

Dissertations:

Frank Maurer: Hypermediabasiertes Knowledge Management für verteilte wissensbasierte Systeme (1993)

Gerd Pews: Flexible Unterstützung für Entwurfsprozesse im Kontext von Verwaltungsaufgaben (1999)

Diploma:

Wolfgang Schittko: Strukturüberwachung von semi-formalen Wissensbasen (1994)

Enrik Beste: Evaluierung und Erweiterung von CoMo-Kit anhand des Beispiels Bebauungsplanung,1995

Dellen, B.: Verwaltung von Abhängigkeiten bei der Operationalisierung von konzeptuellen Modellen, 1995

2.2 The Turn to Software Engineering and the Sonderforschungsbereich 501

In 1992, Professor D. Rombach joined the department of Computer Science in Kaiserslautern by accepting a chair. His activities were devoted to software engineering, in particular to experimental software engineering. A major consequence was in 1995 the creation of the DFG-financed Sonderforschungsbereich 501 (Development of Large Systems with Generic Methods). His appearance had in this context also a major influence to the turn of the CoMo-Kit-Project towards software engineering.

As a consequence, CoMo-Kit had to be adapted to the requirements of software engineering. This was done by integrating CoMo-Kit and the MVP - approach *(Multi-View Process Modeling Language)* of Prof. Rombach into MILOS.

Another long term contact was initiated at that time to Charles Petrie, Stanford University. His influence was in particular in the aspects of designing complex configurations and plans.

Projects and Events:

In this period there were two projects in the Sonderforschungsbereich initiated:

1) Flexible Planung und Steuerung von Softwareentwicklungsprozessen (Richter/Maurer); extending CoMO-Kit towards software engineering processes. After Dr Maurer’s move to the University of Calgary, this project was continued in the next two 3-year periods of the Sonderforschungsbereich.

2) Entwicklung einer flexiblen Modellierungs- und Ausführungsumgebung für Software-Entwicklungsprozesse (Richter/Rombach); integrating MVP and CoMo-Kit.

Theses:

Diploma:

Kirstin Kohler: Design Rationale bei der Modellierung und Abwicklung von Entwurfsprozessen (1995)

Boris Kötting: Konzeption und Implementierung einer Sprache für Software-Entwicklungsprozesse(1996)

Volkmar Pipek: Dynamische Änderungen an Prozeßmodellen, (1996)

Sigrid Goldmann: Procura: A Model of Combined Design Support and Project Management in an Agent-Based Environment (1997)

Thomas. Rüger: Behandlung von Risiken in CoMo-Kit (1998)

Two Locations: Kaiserslautern & Calgary

In 1997, Frank Maurer accepted a professorship at the department of Computer Science, University of Calgary. He has built up his research group since 1997 and his major topic is agile software engineering for web-based systems, extending his previous work centered on the ideas of MILOS. In this time, the work in the Sonderforschungsbereich 501 continued and Frank Maurer was and is still an associated member and his group participated actively in the projects. The two locations concentrated their work on the following views (of course, not entirely):

University of Calgary: Design and implementation of workflow and enactment components of the MILOS system.

University of Kaiserslautern: Concepts for change management, coordination of management/planning activities and their implementation.

In the Sonderforschungsbereich in this time also an internal evaluation of MILOS took place and showed several advantages of MILOS over the present state of the field.

In both locations, initial contacts to industry have been initiated.

In order to continue joint work on MILOS at the now two locations, exchange between the two locations was organized. From Kaiserslautern Sascha Schmitt (1998), Barbara Dellen (1997/98), Michael Schlindwein (1998), Alexander Herzlinger (1999), Boris Kötting (1999, 2000), Martin Schaaf (2000), Jörg Hohwiller (2001) and Harald Holz (1999, 2001) were invited each for several month to Calgary. Barbara Dellen, Martin Schaaf and Harald Holz taught software engineering courses at The University of Calgary. Frank Maurer visited Kaiserslautern each year.

From Calgary, two students Jeremiah Wittevrongel and Phillip Nour (2000) were visiting Kaiserslautern. Starting on August 1st, 1999 a support in the form of travel grants for researchers and students from Kaiserslautern was approved by the DLR

(Deutsches Zentrum für Luft- und Raumfahrt). The complete list of visitors funded by this grant is:

06.02.-14.03.1999 Boris Kötting

20.11.–11.12.1999 Harald Holz, Martin Schaaf

29.04.-20.05.2000 Boris Kötting

31.07.-24.08.2000 Michael M. Richter

04.12.-18.12.2000 Martin Schaaf, Sascha Schmitt

05.05.-26.05.2001 Boris Kötting

25.07.-28.08.2001 Michael M. Richter

01.11.-08.12.2001 Harald Holz

03.08.-31.08.2002 Michael M.Richter

In this period Frank Maurer was, due to the cooperation, co-supervising most of diploma and doctoral theses in Kaiserslautern. In addition, the many joined publications show how fruitful the cooperation was.

Doctoral Dissertations:

Barbara Dellen: Change Impact Analysis Support for Software Development Processes (2000).

Boris Kötting: Der Nutzen virtueller Marktplätze für die Software-Entwicklungsdomäne (2002).

Harald Holz: Process-Based Knowledge Management Support for Software Engineering (2002)

Diploma (Kaiserslautern) :

Christoph Arenz: Modellierung von Softwareengineering-Prozessen mit MILOS (1998)

Sascha Schmitt: Web-based Software Process Modeling with CoMo-Kit (1998)

Thomas Rüger: Behandlung von Risiken in CoMo-Kit (1998)

Frank Neblung: Validierung und Erweiterung der Prozeßmodellierungssprache MILOS (1998)

Ursula Balg: Wiederverwendung von Entwurfsmustern durch ein Workflow-Management-System (1998)

Michael Schlindwein: A Framework Using Computer Agents to Provide Automatic Execution Support for Processes in Software Development Projects (1999)

Andreas Raquet: Dynamic Project Management for Distributed Construction Processes, 1999

Alexander Herzlinger: Distrubuted Workflow Management Approach for Process oriented Management Systems (1999)

Sascha Mungenas: Entwicklung eines Ansatzes zur verteilten Abarbeitung eines Workflows in einem Virtuellen Unternehmen (1999)

Arne Könnecker: Extending a process-centred SEE by context-specific knowledge delivery (2000)

Stella Christofidelli: Verwaltung von Abhängigkeiten und Entscheidungen bei der Planung und Ausführung von Software-Entwicklungsprozessen (2000)

Gerhard Landeck: Modellierung und Verarbeitung domänenspezifischer Abhängigkeiten in MILOS (2001)

Boris Schadt: Die Erweiterung eines virtuellen Marktplatzes zur Unterstützung verteilt operierender Unternehmen (2001)

Jörg Hohwiller: Automatic negotiation on software development tasks (2001)

Master's Thesis (Calgary):

Dale Couprie: Automated Support for a Customized Personal Software Development Process, (1998)

Quan Li: Domain Specific Dependencies (1998)

James Tam: Change Management Support (2000)

Carmen Zannier: Tool support for complex refactoring to design patterns (2003)

Philip Nour: Ontology-based retrieval of software engineering experiences (2003)

Seth Bowen: Peer-to-peer process support (ongoing)

Thomas Chau: Lightweight knowledge management for agile software teams (ongoing)

Kris Read: Tool support for scaling agile teams (ongoing)

Projects and Events:

Visit of Sigrid Goldmann and Harald Holz at the University of Stanford (Dr. Petrie, Prof. Cutkosky, Center for DesignResearch), 20.6.-26.6.1998.

1st Workshop on COORDINATING DISTRIBUTED SOFTWARE DEVELOPMENT on the Eighth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '98). 17-19 June 1998. Stanford University, California, USA. Organized by, Sigrid Goldmann, Harald Holz, Boris Kötting, Frank Maurer.

2nd Workshop on COORDINATING DISTRIBUTED SOFTWARE DEVELOPMENT on the Eighth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '99). 16-18 June 1999, Stanford University, California, USA. Organized by Fawsy Bendeck, Barbara Dellen, Sigrid Goldmann, Harald Holz, Boris Kötting, Frank Maurer.

"International Conference on Software Engineering" (ICSE) 1999: Organization of the Workshop "Software Engineering over the Internet"; organized by Frank Maurer, Mikio Aoyama, Peter Rösch, Richard Webby.

"International Conference on Software Engineering" (ICSE) 2000: Organization of the Workshop "Software Engineering over the Internet"; organized by Frank Maurer, Barbara Dellen, Boris Kötting Israel Ben-Shaul, Stephen E. Dossick, John C. Grundy.

"International Conference on Software Engineering" (ICSE) 2001: Organization of the Workshop "Software Engineering over the Internet"; organized by Frank Maurer, Barbara Dellen, Boris Kötting, John C. Grundy

"International Conference on Software Engineering" (ICSE) 2002: Organization of the Workshop "Global Software Development"; organized by Daniela Damian, Frank Maurer, Nigamanth Sridhar.

Sonderforschungsbereich 501: Workshop on "Software Reuse – Requirements, Technologies and Applications" Kaiserslautern, 10. – 11.3. 2001. Lecture by M.M. Richter: "Similarity and Utility ".

Invited Talks:

S. Goldmann: Internetbasierte Prozessmodellierung und Projektplanung Wissensmanagement 1999. Strategien der Gegenwart, Industriekonferenz, Frankfurt, 1998

B. Dellen: Internetbasiertes Wissensmanagement in Softwareprojekten. Wissensmanagement: Strategien der Gegenwart, Industriekonferenz, Frankfurt, 1999.

M. M. Richter: Knowledge Management als zentrale Unternehmensaufgabe. Customer Relationship Management, Kongress, Düsseldorf, 1999.

M. M. Richter: Flexibles Workflow Management in Software-Engineering Prozessen. Forschungsforum „Wissensmanagement und schnelle Produktentwicklung, Methoden-Trends-Vorgehensweisen". Stuttgart, 1999.

F. Maurer: Agile Methods. APWeb 2001 4th Asia Pacific International Web Conference (Nov. 21-23, 2001, Changsha, P. R. China)

3 The Scientific View: Goals, Concepts, Methods, Architectures, Implementations and Results

In the development of MILOS, one can observe not only views that change naturally over time but also a symbiosis of ideas from software engineering and artificial intelligence. The contributions from AI came mainly from the area of action planning, dependency tracking and knowledge management. It has to be observed, however, that in some areas the methods and even the views had to be extended essentially. Software development was no longer a synthesis of programs in a logical sense and action planning was no longer a completely specified task that could not be changed during the solution phase.

3.1 CoMo-Kit and Semiformal Systems

Initially Como-Kit was intended as a knowledge engineering tool in order to speed up the (iterative) development of distributed knowledge-based systems, in particular systems for fault diagnosis. The essential concept was the notion of a *semiformal system*. Semiformal means that parts are formally understandable by a formal system and other parts need the interpretation of a human. In this sense, a semiformal system is a kind of an assistant system that helps humans to accomplish their tasks. This view was a novelty in the area of knowledge based systems at this time and became a leading view later on in MILOS. It also is now a focus in the discussion of lightweight knowledge management approaches.

The basic representation structure was a graph, where the nodes represented objects like tasks or products and the edges were labeled by actions. If the labeling was formal, a formal execution could be performed. Informal labels represented hypertext/hypermedia entities.

The graphical modeling tool allowed the decomposition of a goal into subgoals that are independently solved.

Techniques, methods and systems were developed which supported planning and enacting complex distributed cooperative design processes. The essential notions are tasks, methods, products (concepts) and agents. Conceptually, the definition and use of various kinds of dependencies played a major role: in particular the causal dependencies between input and output products.

Enactment support was provided in form of a modeling tool and a workflow management system in the form of a scheduler. The implementation of the latter one was based on the general planning and design system *Redux*, developed in Stanford by Charles Petrie. Redux can be seen as a structured justification based truth-maintenance system (JTMS); it makes use of special dependencies in the form of justifications. Redux was later on essentially used in MILOS.

The project in Urban Land Planning provided just those problems which were suitable for CoMo-Kit. Such planning may last over years and hence planning and enactment are interleaved. In this time the context as well as the intentions of administrations and involved people are up to change and hence a high degree of flexibility is required.

As indicated, many of the key ideas of the MILOS approach were already being discussed in CoMo-Kit occur. In Como-Kit, however, there have been many restrictions, both to the concepts and even more to the implementations (although they are not mentioned here in detail), which put boundaries on the applicability, which were in the sequel stepwise removed.

3.2 Software Engineering: CoMo-Kit and MVP and subsequent extensions of MILOS.

In 1995, the challenges concerning software engineering came up, mainly by the Sonderforschungsbereich 501. A major influence from the side of Professor Rombach came not only in the form of a technical contribution. It was the whole view on software engineering that this area is not the discipline where logically correct algorithms are specified and designed but as a practical construction of useful artifacts.

In this view, software engineering is a discipline where experiments, measurements, experiences and approaximations have their natural place. This view was a basis that made cooperation with an approach using semiformal systems possible and it was the basic view of the whole Sonderforschungsbereich.

There were two major activities:

- Extension/Adaptation of CoMo-Kit for SE-processes
- Cooperation with the group of Prof. Rombach in order to define the modeling language MILOS

For the integration of the approaches of CoMo-Kit *(Conceptual Modeling Kit)* and MVP which had MVP-L *(Multi-View Process Modeling Language)* the essential elements where joined. The situation was as follows:

CoMo-Kit was intended to support dynamic planning and enactment of specific technical processes that are characterized by an inherent creativity and therefore require a high flexibility in planning and execution. CoMo-Kit provided a process engine and an approach for mechanisms to control dependencies between products and processes.

In the scope of the MVP-Project, an approach to support processes, planning, execution, simulation, improvement and reuse on the basis of software engineering principles was developed. The process modeling language MVP-L *(Multi-View Process Modeling Language)* was used. In MVP-L project plans can be formulated which are instantiated in the run-time environment MVP-S. This approach supports process modeling in the large.

Comparison of the two approaches from a technical point of view:

The strengths of CoMo-Kit are its flexibility which allows to support creative processes and, as a consequence, replanning and management between elements of a process. MVP on the other hand provides powerful modeling concepts, methods for static and dynamic analysis and the use of measurement for process enactment.

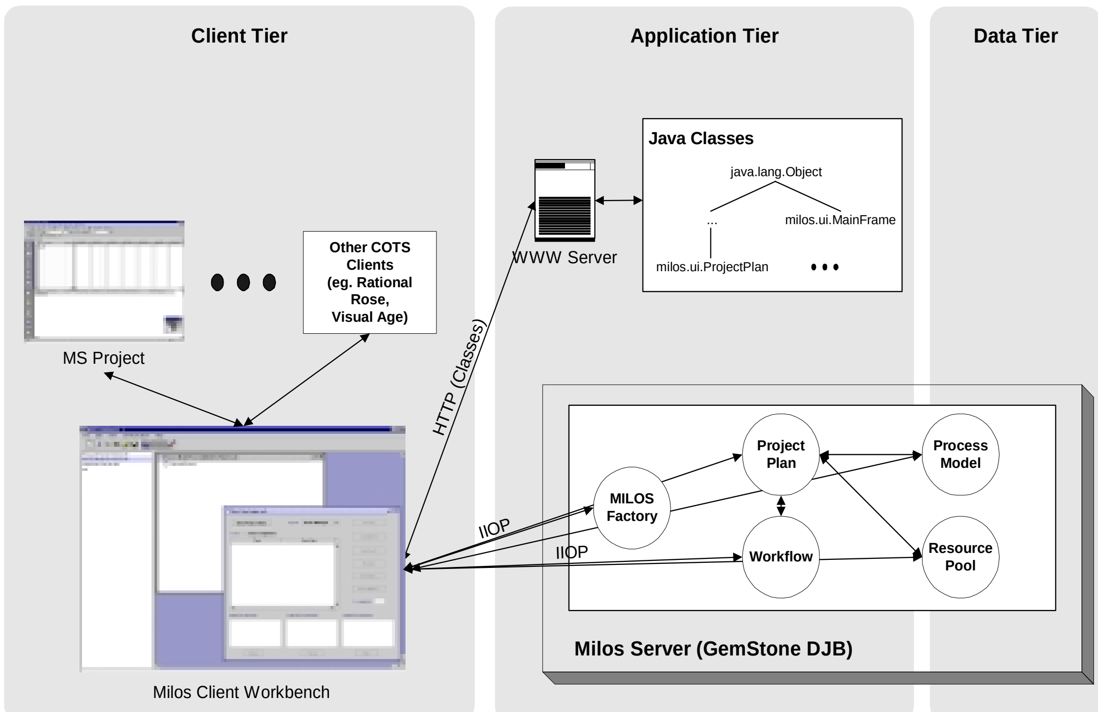
The basic integration problem was to combine the flexibility of CoMo-Kit with the expressiveness of MVP-L. As a consequence, CoMo-Kit was adapted to the needs of software engineering.

The first step was the development of an integrated process modeling language. An initial language concept for an integrated process support approach integrating MVP and CoMo-Kit was presented under the name of MILOS *(Modeling Language and Operational Support for Software Processes).*

The prototype was implemented in Java and is based on a „three-tiered“ architecture:

Client Tier – Application Tier – Data Tier.

The clients are designed as Java Applets. The server components are integrated in the object oriented DBMS GemStone which is responsible for persistence- and transaction services.



The MILOS - GemStone application model

The next step was the specification of some set of operational concepts.

The long time goal of MILOS was formulated as the development of methods and techniques for flexible planning and control of complex software development processes. Major tasks in this context were:

- Increasing the traceability of decisions in the planning phase as well as in the enactment phase of software development processes.
- Incremental refinement of (incomplete) project plans during the project enactment using project specific knowledge.
- The ability to change project plans „on the fly“ as a reaction to changing context conditions, assumptions and planning errors.
- Treatment of effects of changes, in particular the notification of team members affected by the changes and management of team and individual processes.
- Allow the definition of design rationales and develop mechanisms for their control.

A major role for these purposes played the development of techniques for formulating, acquiring, inferring and controlling various kinds of dependencies and dependency patterns. These tasks were not only realized in the sequel but also extended in several aspects.

In a case study in the Sonderforschungsbereich an internal evaluation of MILOS took place. The results showed an advantage of MILOS with respect to notification support for complex replanning, a more efficient manual notification and a substantial general effort reduction when MILOS was used.

After the two locations had been established several of the extensions of MILOS were jointly achieved. After some time different aspects for research directions were formulated in both locations, which resulted in different directions for the system development:

- In Kaiserslautern emphasis was put on refinements of the original system; the extensions concentrated on meta planning and process oriented knowledge management (see 3.4.4 and 3.4.5).
- In Calgary emphasis was put on increasing the applicability of the approach, and redesigning the system, based on a new architecture (Java Enterprise Beans). Furthermore, the tools was adapted to support agile development teams (see 3.5)

On the conceptual level results were exchanged but the source code was developed separately.

3.3 Concepts and Structures of MILOS

In the following, we will describe the main modeling concepts and structures developed by the MILOS project. Almost all of them were realized in some versions of the MILOS implementation but not all of them are present in the most recent implementations.

3.3.1 The Process Modeling Language of MILOS

Systems that support planning and coordination of software development have to represent the entities used in project plans. MILOS allows representing knowledge about software development processes. MILOS allows defining

- Process, product and resource models
- Project plans
- Product deliverables developed within projects
- Background knowledge such as guidelines, business rules, empirical studies etc.

The *process model* component handles process definitions including control flow (pre- and post conditions), information flow, and process decompositions. Process models can be seen as a type system for project plans. For every process, required skills may be described.

Product and resource models can be developed and integrated into process models. Product models define the type of information that a task uses or produces. Resource models describe the skills of agents (team members) carrying out development tasks and the team structure.

For a new project, these process models can be the basis for the definition of project plans.

3.3.2 Product Models

MILOS allows the specification of hierarchical, object-oriented product models. A *product type* defines the structure of a set of products with the same behavior. A product type is either basic or complex. Basic types are predefined and may be integer, real, string, text, date, or external references such as a www page URLs or word documents. A complex type consists of one or more typed subproducts and attributes. Complex product types can be specialized (IS-A relation).

Based on a given product model, products instances (synonyms: products, deliverables) that contain general and specific project knowledge of a company can be defined.

3.3.3 Process Models and Methods

Within process models, activities and their interrelationship are described.

A process is defined by a description of the process goal, a set of conditions, process attributes, the products needed to plan and to execute the process, a set of alternative methods to reach the process goal, the products to be produced and resource allocations.

Methods are either *complex* or *atomic*. Complex methods refine a process into one or more subprocesses whereas the application of an atomic method results in the production of products that are the outputs of the process.

Inputs of a process may either be products that are produced by other processes during project enactment or predefined background knowledge (e.g. manuals or guidelines) taken from generic process models.

Process models are generic descriptions of the general way of the course of projects. The models have to be specialized and customized within the projects.

Every process is associated with a set of roles and qualifications which are needed to perform the task (e.g. the process „Implement Class“ is associated with the qualifications „Java knowledge available“ and the role „Programmer“).

3.3.4 Resource Models

The *resource pool* component manages roles, agents and agent properties. It allows representing the organizational structure of a company as well as a skill ontology. An agent can have a set of skills. Resource models allow assigning roles and qualifications to project team members. These models describe knowledge needed by project managers to find appropriate people for a task. The project planner can query the resource pool for all agents with specific skills.

3.3.5 Overview: Table of MILOS's representation primitives

Concept	Description
process	Set of activities that have to be executed in order to reach a given goal.
condition	A condition controls the execution of a project plan. We distinguish between preconditions and postconditions.
product type	The description of type and structure of a class of products.
product	A product is an information unit of a given product type, for example a document or a piece of code.
product reference	Product references stand for the type of products that are used and/or produced by a process or a method. We distinguish between products that are consumed for planning, consumed for execution, produced, and modified.
produce	This parameter type stands for a product of a given type that is produced by an atomic method of a process.
consume for execution	Product reference that describes the read only access to a product during the execution of a process.
product mapping	Defines the product flow within the process/method hierarchy.
method	Problem solving method to reach a process goal.
atomic method	Leaf within the process/method hierarchy. It produces or modifies a product.
complex method	Problem solving method that refines a process into one or more subprocesses.
attribute	Attributes are properties of processes, products, methods and resources.
product attribute	Attribute of a product.
process attribute	Attribute of a process.
resource	Resources are used for the execution of the project. We distinguish between two types of resources: agents and tools.
resource attribute	Attribute of a resource.
role	A predefined resource attribute. It describes the task of a resource within an organization.
qualification	A predefined resource attribute. It describes a skill of a resource.
tool	A program that supports activities.
agent	A human or machine that carries out activities within a process.
precondition	A condition that has to be valid to carry out a process.

postcondition	A condition that has to be valid after a process has been executed.

3.3.6 A Key Structure: Dependencies

Actions are not isolated but have various dependencies of a complex character. Many dependencies are „silent“ and activated by certain events. In actual situations dependencies have to be identified and put into actions. Management requires a thorough understanding of these relations in order to perform optimal activities and to structure the dependencies.

The representation of dynamic dependencies has to take into account that changes lead dynamically to activities. All dynamic activities are governed by ECA rules.

An ECA rule has the structure:

- events: changes to the project plan/state
- conditions: Describe the circumstances under which the event triggers the action
- actions: e.g. notifications

In more detail:

In MILOS events trigger notifications. E.g. a project plan causes an event if the planer adds a new process to the plan. As a consequence, the workflow engine will be informed.

There are two kinds of events in MILOS:

Complex events combine different events, e.g.
MILOSProcessDefinitionChangedEvent

Basic events represent exactly one precisely defined event as e.g.
MethodAddedEvent.

A change rule references an action class and a condition. The action class describes an executable action (e.g.. NotifyAction)

The condition describes when the action will be executed after the event occurred.

In principle there are two types of dependencies:

- Those derived from a process.
- Those specific for the application domain.

The *project plan management* component allows customizing a process model, resulting in a project plan. Beside adding/removing tasks, customizing includes scheduling planned start and end times of processes and assigning agents to processes.

The *workflow management* component is responsible for enacting the project plan and managing products. It generates to-do lists for agents and maintains the current state of the project. The workflow engine reacts dynamically to project plan changes during execution, without interrupting the execution.

Process modeling, planning and execution can be distributed to different companies and areas by running the clients on an arbitrary machine connected to the Internet.

3.3.8 Project planning

Project planning is essentially an instantiation of process models. Planning a project comprises

- selecting appropriate processes from the experience base, and inserting them into the plan,
- selecting applicable development methods according to the characteristics of the organizations (e.g. familiarity with specific methods) and the goals of the project (e.g. budget limitations),
- instantiating variables in pre- and postconditions,
- allocating resources to the processes according to the resource properties specified in the respective process models, and time scheduling with these resource allocations in mind.

3.3.9 Project Enactment

Once an initial plan has been built, it can be executed: Agents work on processes assigned to them, thereby creating the corresponding output products.

As specified in the model, certain conditions must be fulfilled before a process can be started. These conditions can either be always true, or they become true when a certain stage in the project is reached, e.g. when all of a process' inputs are complete and meet the specified quality requirements.

3.3.10 Interleaving

Project planning, project enactment and execution can be interleaved. This is the major point requiring flexibility.

3.3.11 Workflows in MILOS

A process model in MILOS consists of a set of process types and decomposition relationships between these types. A project plan consists of a set of project instances and decomposition relationships between them. The MILOS workflow engine interprets project plans in order to guide (human) users.

Local workflows on one machine in MILOS are based on a workflow kernel which uses an Object-Version-Management (OVM). It stores product version instances and associates parameters to instances. In order to handle changes in the process model as well as in the process plan MILOS uses a special event concept.

The extension to *distributed workflows* has to assume different environments (domains) with partial workflows, each in one specific domain. One domain is declared as leading and from this domain the communication takes place via domain interfaces. It is possible that the workflow of some domain is only partially visible in the leading domain. Parts of the tasks of the OVM are integrity checks between the different workflows. This made it possible that each engine can work off-line independent of the existence of a server workflow engine.

It should also be mentioned that workflows are not static but also up to change.

3.4 Overview over the MILOS-Systems

Due to changing research directions of both groups involved in implementing the concepts underlying the MILOS approach, it was decided in 1999 to split up into two development streams (as indicated above):

MILOS-KL, developed in Germany, focuses on virtual companies, process-oriented knowledge management, and meta-planning.

MILOS-ASE (or M-ASE), developed in Canada, focuses on agile methods support, web-based system development and knowledge management by building communities of practice

Both approaches rely on similar modeling concepts. In the following, we provide a short overview on these.

3.4.1 The Main Present Functionalities of MILOS-KL

Internet-based project coordinationProcess modeling, project planning and project enactment in one system:

MILOS modeling tools

Interface to external tools

XML import/export interface

Possibility to use existing reference models

Process modeling:

- Explicitly define the software process
- Adaptable to specific project needs

Various levels:

- Life-cycle models
- Project plans

Project planning:

- MILOS plannings-GUIs
- Interface to MS Project
- MS-Office, FrameMaker and RationalRose can be invoked from within the MILOS execution framework.
- A project planner can create an initial plan using some standard software like MS-Project.
- The project planner can import the plan from MS-Project into MILOS. And then team members, regardless of their geographical location, can log into

MILOS via standard Web browsers and are provided with individual workspaces.

Project enactment: Execution support:

Automation of To-do lists

Feedback:

Forecasted end date

Current task state

Access to input products and output products

Change Support, Notifications

Plan changes during execution:

Replanning triggers automatic update of the execution state

The agents can update their own work schedule; in particular their forecasts on start and finish times.

Storage and retrieval of decision contexts

Automated updates of project states

Establishment and change of output products

Notification of involved persons

Different kinds of notification techniques are supported: standard notifications like escalation mechanisms (e.g. user notification on approaching deadlines), notification dependencies from the project plan, and project participants can express interest in specific information.

An extension facilitates change management based on UML-product-specific knowledge; it is not anymore restricted to strictly typed MILOS entities

Project-Trace for subsequent analysis:

Recording events during project enactment

Storage in XML file

Graphical representation in MILOS

Reuse of decisions from former projects

Dependency Management: Management of change rules:

Definition of change rules: Heuristics

Mappings: Dependencies of change rules

Propagation of change effects

The event notification component automatically detects dependencies resulting from data and control flow and provides the right information to the right people at the right time. *product-changed* or *process-delayed.*

The MILOS-KL environment consists of several components: a resource pool, a process modeling component, a project management component, and a workflow engine. These components are linked by a change management mechanism.

MILOS-KL offers several language concepts for defining measurement based process models, object oriented product models and resource models.

MILOS Project-Trace:

This extension allows to store both changes of project plans and states of the processes and products persistently. The set of data generated here automatically is seen as a project trace. The trace is available in XML.

3*.4.2 Support of Organizational Learning (see also 3.4.4):*

MILOS supports the feedback cycle for creating learning organizations in several ways:

- Uploading project plans to an experience base
- Adding new INOs (see 3.4.4)
- Supporting specific discussion and news groups
- Creating FAQs
- Evaluating information usefulness
- Supporting learning in virtual teams

MILOS also supports learning from experience:

- Information needs are represented as queries
- The information assistant presents queries as INOs (see 3.4.4) and information sources
- Context sensitive similarity based queries (using multiple similarity measures) are allowed
- As a result information about similar processes is provided.

3.4.3 Virtual Enterprises and Virtual Market Places

Important for virtual enterprises is the question of data security. This was realized by the extensions of MILOS concerning distributed workflows. Motivated by this research a Virtual Marketplace System (VMS) was realized (Boris Kötting). The purpose of this marketplace was to support outsourcing of partial projects of software development processes. It is a comprehensive system which offers not only a great variety of possibilities for auctions but also supports all necessary communications and activities. Partially the activities have been automated and partially the worked as an assistant system.

3.4.4 Management Activities and their Coordination

Due to the fact that MILOS deals with tasks of incomplete and changing information the management of knowledge and information naturally plays an important role. As a major result, a detailed life-cycle model for Software Engineering Process-Oriented Knowledge Management (SE-POKM, Harald Holz) was established. Process-orientation is an important factor for the successful introduction of knowledge management support into organizations as it builds on accepted concepts on Business Process Modeling (BPM). Central concepts are:

Information Need: Concerned with information not provided by the process definition; this depends context and experience.

Information Source: Electronic source from which the information can be obtained.

Information Need Object (INO): They enrich the process model and support both, process planning and process execution.

The three major contributions of the SE-POKM model are:

- An explicit representation of dynamic (situation-specific) information needs
- A specification of the information need retrieval
- A guideline for the experience packaging phase

In order to illustrate and facilitate the adoption of the SE-POKM model, MILOS has been extended in order to provide corresponding tool support (see below).

The Knowledge Department (KD) has a fine-granular focus on the individual activities: for each process type, it tries to elicit what information resources might be useful for agents involved in activities of that type. A feedback loop has to be established, aiming at an organizational learning process.

Because of the KD's focus on individual activities, the process model might need to be refined during the elicitation of information resources. Likewise, changes to the process model generally trigger changes to the information resource model.

3.4.5 Meta Tasks

Planning is oriented at a process model and can take place on different levels:

Meta level: Deals with meta tasks

Basic level: Direct planning tasks.

Up to now in MILOS only the basic level was considered. Meta tasks have been present only implicitly and were performed on the basic level. In very large and distributed situations this view is very restricted, in particular in change management. The extensions are concerned with

Explicit representation and management of planning process

Representing planning tasks explicitly as *meta tasks* (which manipulate the plan, in contrast to *object level processes*, which create products)

Automatic as well as user-defined generation rules to create meta tasks

In a meta plan the dependencies between activities are modeled, i.e. each activity is modeled as a meta task in a plan whose execution again results in a (project) plan.

In the definition of meta tasks rules are included for their automatic generation and delegation. Situations are to be identified in which the user might need to define his/her own meta tasks. The MILOS process modeling language is extended for that purpose.

Each meta task can be supported by a knowledge management component, i.e. the KM can proactively provide the planner with information on how best to execute this particular task.

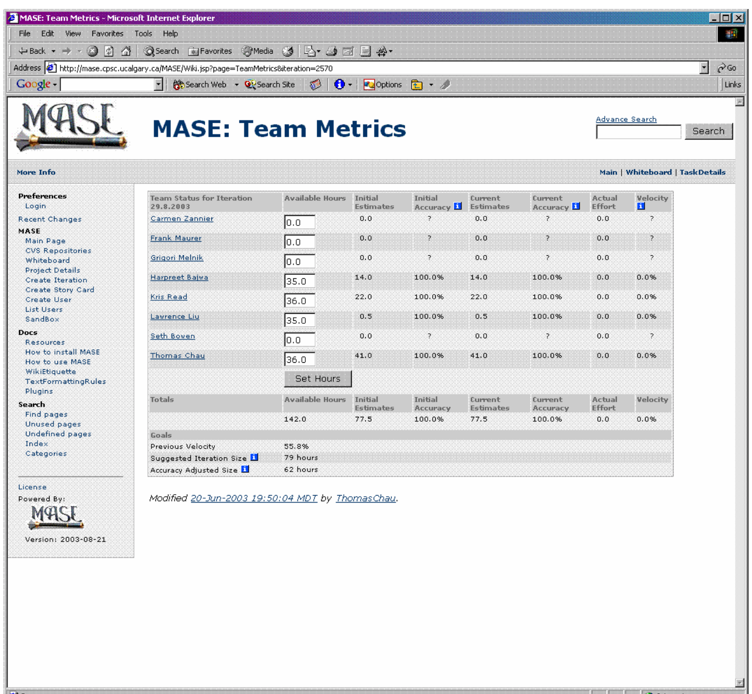
3.5 An Overview over M-ASE

The MILOS-ASE (or M-ASE) project investigates methods, techniques, and tools that support the coordination of virtual software engineering projects using agile methods. Key areas of interest are:

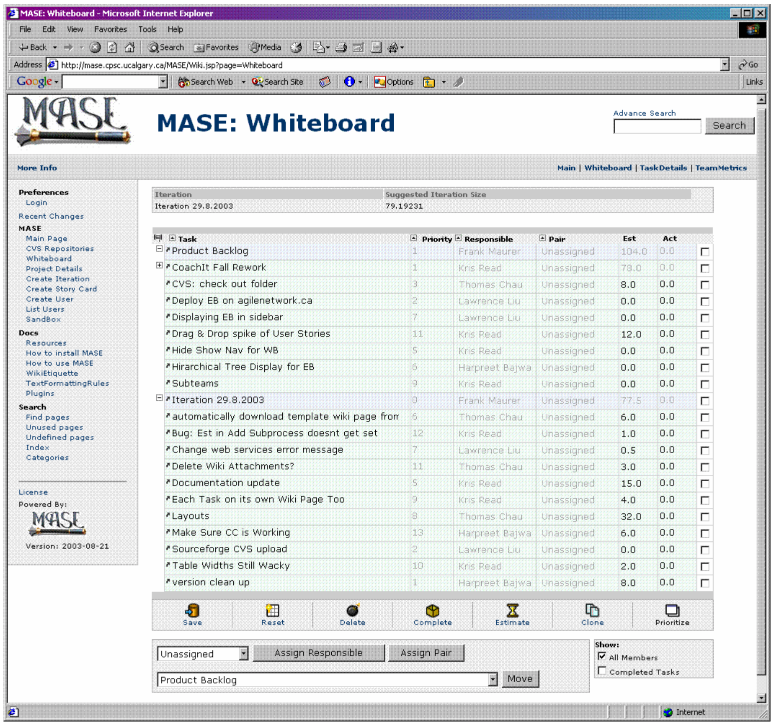
- Agile practices for distributed teams
- Internet-based Software Process Support
- Virtual Software Enterprises
- Experience & Knowledge Management

To adopt MILOS towards the culture and way to work of agile software teams, a first step was to reduce the MILOS approach to its bare bones: task definition and progress tracking.

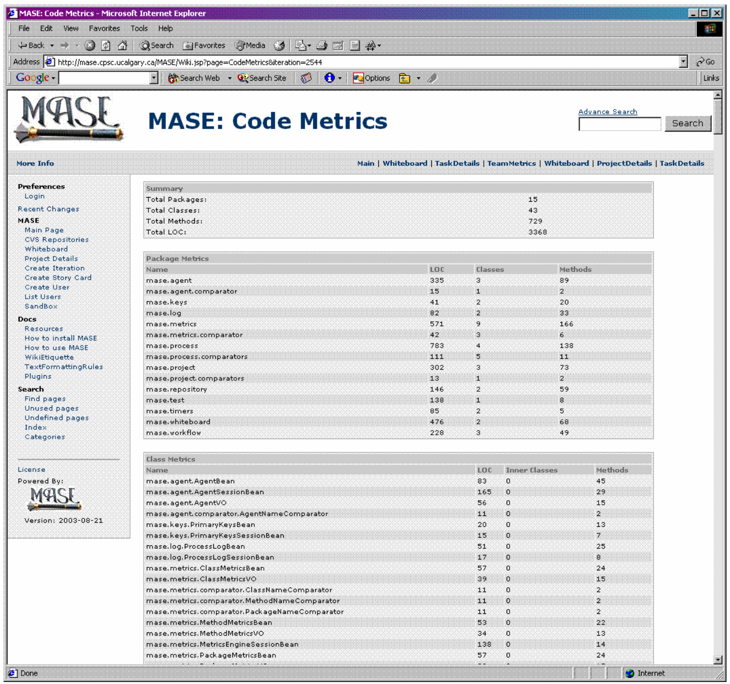
Project planning starts with the team deciding how many work hours can go into the next iteration. Based on this information and the velocity of the last iteration, MASE is able to suggest the size of the next iteration.



The team then uses the planning whiteboard to move tasks from the product backlog into the next iteration. Team members who take responsibility of a task estimate the effort. If the sum of all efforts is higher than the suggested iteration size, the customer decides which tasks are postponed.



For each iteration, MASE is able to keep track on various source code metrics (including LOC, number of classes and methods, cyclomatic complexity). In addition, the system keeps track of several test metrics (#JUnit tests and broken builds) by integrating with CruiseControl.



The system is also keeping track of test metrics, initial estimates, current estimates and actual effort.

3.6 Design and Implementation of Components and Additions of the MILOS System.

Here we discuss some major extensions of MILOS explicitly.

3.6.1 VMSDT (Virtual Market Place for Software Development Tasks, see 3.4.3)

In the implementation the Virtual Market Place was connected to MILOS: MILOS supported outsourcing of subprojects down to the level of workflows. A task consists of the following parameters: Author information, addressee list, goal description, schedule, estimated effort, payment conditions, input/output parameters, required skills, offering deadlines, child tasks (for complex tasks), contract information and negotiation protocol.

3.6.2 PRIME (Process-Oriented Information Resource Management Environment, see 3.4.2)

PRIME is an implementation of the retrieval of the information needs formulated in the SE-POKM model. Its purpose is to provide an agent with a set of information resources considered to be useful by the process group for performing activities. It provides a technical infrastructure for knowledge distribution and feedback communication.

The central component of PRIME is the Information Resource Manager. He is connected with several components dealing with the characterization of activities and related entities. The components are a knowledge base, a modeler, an updater, an editor and a manager. The Information Resource Manager has furthermore a knowledge base and access to an Information resource retriever component as well as to an Information Source Manager.

There are also some auxiliary assistant components. Important is the Information Assistant that realizes a task-sensitive information delivery concept. Monitoring and evolution is supported actively by PRIME via the Information Request and Feedback Forum (IRFF) component. Any communication between agents and the knowledge department is maintained for later analysis with the objective to update the information resources model in such a way that the Information Assistant will provide agents with all useful information during future activities.

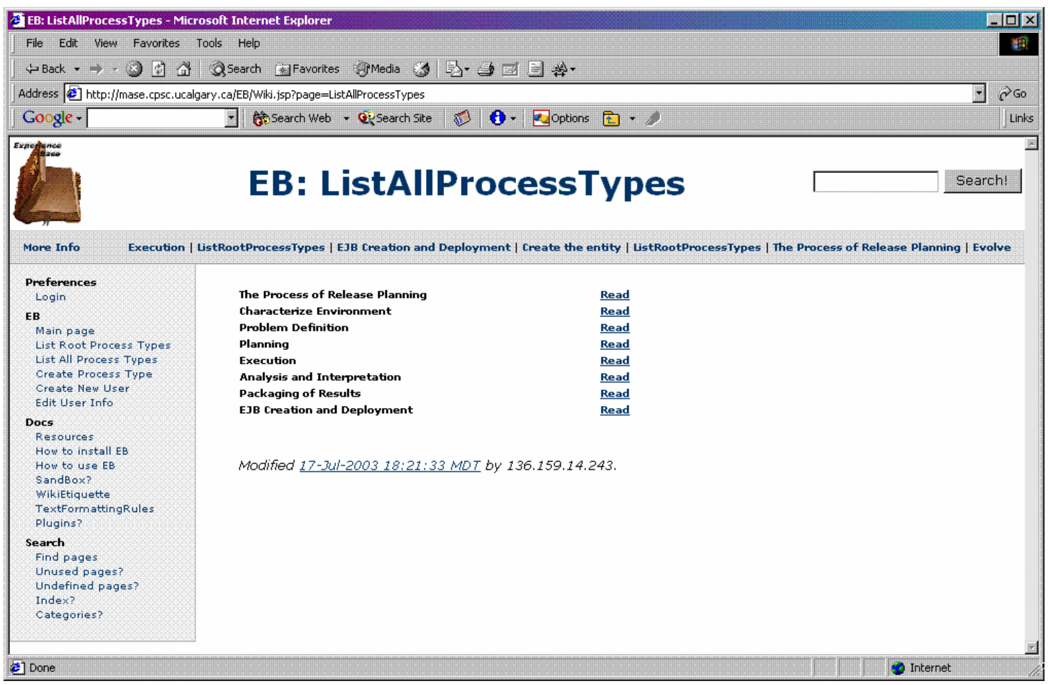
PRIME is integrated into MILOS-KL environment such that it can provide users with activity- and situation-specific information from information sources available to the MILOS environment.

3.6.3. COACH-IT

Agile software development processes are best applied to small teams on small to medium sized projects. Scaling agile methodologies is desired in order to bring the benefits of agile to larger, more complex projects. One way to scale agile methods is via an architecture-centric approach, in which a project is divided into smaller modules on which sub teams can use agile effectively. However, a problem with architecture-centric modifications to agile methods is the introduction of non-agile elements, for instance up-front design and integration difficulties. The COACH-IT extension of MASE allows to define the architecture of a software system and enforces the contracts between components by executing test drivers.

3.6.4. Lightweight Knowledge Management

Traditional approaches to learning software organizations (including PRIME) rely on the externalization and structuring of knowledge while agile approaches rely on sharing of tacit knowledge among team members. To overcome this discrepancy, MASE integrates structured and unstructured knowledge sharing tools. On the one hand, MASE allows several project teams to share a process centred experience base. On the other hand, users are able to share knowledge by externalizing it into Wiki pages. Both, the MASE Experience Base as well as the MASE Project Support System use Wiki technology. Thus, knowledge sharing within a team as well as across team boundaries is supported. In addition, MASE indexes the Wiki pages as well as the database contents in the same text retrieval engine. Thus, a user is able to search for information in a single query that retrieves textual information as well as database contents.



4. Applications and Relations to Industry; Influence on Teaching

4.1 The first application (using CoMo-Kit) was the design and implementation of a system supporting the development of a system for Urban Land-Use Planning

(Intelligente Bebauungsplannung, IBP). It was financed by the VW-Foundation and carried out in cooperation with Prof. Streich from the department of architecture and city development at the university of Kaiserslautern who gave advice on domain knowledge. Partially there was a cooperation with the city of Kaiserslautern.

4.2 In Calgary there was a corporation with NORTEL, in particular with respect to change management.

4.3 In 2001/2001 MILOS was evaluated by the Daimler-Chrysler Company for their specific needs.

4.4 In Calgary, MASE is currently used by a small company (Wireless Edge) for supporting the collaboration between customer and development team.

4.5 Teaching**:** In Calgary, MASE is now be used for undergraduate as well as graduate teaching.

4.6 Teaching: In summer 2003 PRIME was used in Calgary in a graduate course on Knowledge Management.

Publications

The list of publications should be seen in connection with the doctoral, diploma and masters theses, they are not listed here again. The list does not contain internal reports.

The following URLs provide access to most of the publications:

- http://sern.ucalgary.ca/~milos/Library.htm
- http://wwwagse.informatik.uni-kl.de/

1992 – 1995 (CoMo-Kit and Knowledge Engineering):

F. Maurer: HyperCAKE: Ein Wissensakquisitionssystem für hypermediabasierte Expertensysteme, in: J. Biethahn, R. Bogaschewsky, U. Hoppe (Hrsg.): Expertensysteme in der Wirtschaft 1992 - Anwendungen und Integration mit Hypermedia, pp. 85-106, Gabler Verlag, 1992.

F. Maurer, R. Traphöner: A Knowledge Acquisition Environment for Hypermedia-Based Expert Systems, in: S. Hashemi and J.P. Marciano and G. Gouarderes: Proc. 4th International Conference on Artificial Intelligence & Expert Systems Applications (EXPERSYS-92), Paris, Oct. 21-22, 1992.

F. Maurer, S. Neubert: A Tool for Model-based Knowledge Engineering. Proceedings Avignon'93, 1993.

F. Maurer, G. Pews: Hypermedia als Zwischenrepräsentation bei der Expertensystementwicklung, in: H. P. Frei, P. Schäuble (Eds.): Hypermedia, pp. 93-104, Springer, 1993.

F. Maurer, G. Pews: Validierung von konzeptuellen Modellen, in: F. Puppe, A. Günter (Eds.): Expertensysteme 93, pp. 125-139, Springer, 1993.

F. Maurer, G. Pews: A distributed operationalization of conceptual models, in: Proceedings EKAW-93, pp. 149-164, Toulouse, 1993.

F. Maurer, J. Paulokat: Operationalizing Conceptual Models Based on a Model of Dependencies, in: A. Cohn (Ed.): ECAI 94. 11th European Conference on Artificial Intelligence, pp. 508-515, John Wiley & Sons, Ltd, 1994.

Dellen, B., Maurer, F., Paulokat, J.: Verwaltung von Abhängigkeiten in kooperativen wissensbasierten Arbeitsabläufen, In: Richter/Maurer (Eds.): Expertensysteme 95, infix Verlag, pp.72-89.

F. Maurer, G. Pews: Ein Knowledge Engineering Ansatz für kooperatives Design am Beispiel der Bebauungsplanung, Themenheft Knowledge Engineering KI 1/95, interdata Verlag, 1995.

G. Pews: Ein WWW-Informationssystem für Entwurfsprozesse, Workshop Hypermedia & KI 95

1996 (First concepts for MILOS and further development and applications of CoMoKit):

B. Dellen, K. Kohler, F. Maurer: Integrating Software Process Models and Design Rationales, Proceedings Knowledge-Based Software Engineering KBSE-96, IEEE press, 1996.

B. Dellen, F. Maurer: Integrating Planning and Execution in Software Development Processes, Proceedings WET ICE 96, pp. 170-176, IEEE press, 1996.

Goldmann, S.: Procura: A Project Management Model of Concurrent Planning and Design, Workshop on Project Coordination of the WETICE Conference

F. Maurer: Current Status of Case-Retrieval in Engineering Domains: An Analysis from the Knowledge Engineering Perspective, Knowledge-based System Journal, Vol.9 No. 2, pp. 83-91, 1996.

F. Maurer: Coordinating System Development Processes, Proceedings Knowledge Acquisition Workshop 1996 (KAW-96), Banff, Vol. 2, pages 49/1-49/20, 1996.

F. Maurer: Computer Support in Project Coordination (Workshop Summary), Proceedings WET ICE 96, p. 200-205, IEEE press, 1996.

F. Maurer, G. Pews: Supporting Cooperative Work in Urban Land-Use Planning, Proceedings COOP-96, 1996.

F. Maurer, G. Pews: Flexibles Workflowmanagement für Entwurfsprozesse am Beispiel der Bebauungsplanung, Proceedings CAD-96, pp. 256-270, Carl-Hanser Verlag, 1996.

M. Verlage, B. Dellen, F. Maurer, J. Münch: A synthesis of two software process support approaches, Proceedings 8th Software & Engineering and Knowledge Engineering (SEKE-96), Skokie (IL), USA, June 1996. *(This is paper is on the integration of CoMo-Kit and MVP)*

F. Maurer: Project Coordination in Design Processes, Proceedings WET ICE 96, pages 191-198, IEEE press, 1996.

1997 (Extending CoMoKit for administration of software process models; workflows)

J. Angele, R. Perkuhn, R. Studer, A. Oberweis, G. Zimmermann, F. Maurer, B. Dellen, G. Pews, W. Stein: Abschlussbericht der GI Arbeitsgruppe Vergleichende Analyse von Problemstellungen und Lösungsansätzen in den Fachgebieten Information Systems Engineering, Software Engineering und Knowledge Engineering, EMISA Forum, Heft 2, 1997.

Dellen, B., Holz, Pews, G.: Wissensmanagement in CoMo-Kit, Proc. of the KI-97

B. Dellen, H. Holz, F. Maurer, G. Pews: Wissensbasierte Techniken zur Flexibilisierung von Workflowsystemen . Proc. Software Management 1997, B.G. Teubner Verlagsgesellschaft, Reihe Wirtschaftsinformatik, October 1997.

B. Dellen, F. Maurer, J. Münch, M. Verlage: Enriching Software Process Support by Knowledge-based Techniques, Int. Journal of Software Engineering and Knowledge Engineering, Volume 7, No. 2, pp. 185-215, 1997.

B. Dellen, F. Maurer, and G. Pews: Knowledge Based Techniques to Increase the Flexibility of Workflow Management, Data & Knowledge Engineering Journal, Vol. 23 No. 3, pp. 269-295, September 1997.

F. Maurer: CoMo-Kit: Knowledge Based Workflow Management, Proceedings Workshop on Knowledge Management, AAAI Spring Symposium, pp. 106-110, 1997.

1998 (Implementation, Reimplementation and and Extensions of CoMoKit concepts in MILOS; Reuse)

Bendeck, F.,Goldmann, S., Holz, H., Kötting, B.: Coordinating Management Activities in Distributed Software Development Projects, Proceedings of the 7 th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 33-38, 1998

Dellen, B., Maurer, F.: Change impact analysis support for software development processes, Journal of Applied Software Technology, International Academic Publishing, 1998.

Dellen, B., Maurer, F.: A Concept for an Internet-based Process-Oriented Knowledge Management Environment, Process-Oriented Knowledge Management Environment; Proceedings of the KAW'98, Banff, Canada, 1998

Dellen, B., Maurer, F.: Internet Based Software Process Support, Proceedings of the 7 th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 27-32, 1998

B. Dellen, F. Maurer: Change impact analysis support for software development processes, International Journal of Applied Software Technology, ISSN 1198-5577, Vol 4, No 2/3, International Academic Publishing, Scarborough, Ontario, Canada,

1998.

Raimund L. Feldmann, Jürgen Münch, Stefan Vorwieger: Towards Goal-Oriented Organizational Learning: Representing and Maintaining Knowledge in an Experience Base. In: Proceedings of the Tenth International Conference on Software Engineering and Knowledge Engineering (SEKE‘98), pages 236–245, San Francisco Bay, CA, USA, June 1998.

Raimund L. Feldmann, Stefan Vorwieger: Providing an Experience Base in a research Context via the Internet. In: Proceedings of the ICSE 98 Workshop on "Software Engineering over the Internet",

Birgit Geppert, Frank Rößler, Raimund L. Feldmann, Stefan Vorwieger: Combining SDL Patterns with Continuous Quality Improvement: An Experience Factory Tailored to SDL Patterns. In: Proceedings of the 1st Workshop of the SDL Forum Society on SDL and MSC (SAM98), pp 97–106, Berlin, Germany, 1998. ISSN: 0863-095.

Holz, H., Goldmann, S., Maurer, F.: Working Group Report on Coordinating Distributed Software Development Projects, Proceedings of the 7 th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 69-72, 1998

H. Holz, S. Goldmann, F. Maurer: Coordinating Distributed Software Development Projects, in Proceedings Workshops on Enabling Technologies: Infrastructure for the Collaborative Enterprise (WETICE 98), IEEE press, 1998.

F. Maurer, B. Dellen: Internet Based Software Process Support. Proc.on Software Engineering over the Internet held at the International Software Engineering Conference (ICSE-98), Kyoto, Japan.

F. Maurer, G. Kaiser: Software Engineering in the Internet Age, IEEE Internet Computing Magazine, Vol 2, No 5, September1998.

Petrie, Ch., Goldmann, S., Raquet, A.: Agent-Based Project Management, Proc. of the Internl.Workshop on Intelligent Agents in CSCW, Deutsche Telekom, Dortmund, Sept. 1998, p. 1-1. (short version of a paper with the same title in Springer LNAI 1600)

Entwicklung großer Systeme mit generischen Methoden – Eine Übersicht über den Sonderforschungsbereich 501, 1998.

1999 *(Extension of MILOS towards a distributed system; virtual enterprises, reuse)*

St. Decker, F. Maurer: Editorial: Organizational Memory and Knowledge Management, Special Issue on Organizational Memory and Knowledge Management of the International Journal of Human-Computer Studies, Sep 1999.

Raimund L. Feldmann: Developing a Tailored Reuse Repository Structure – Experience and First Results. In: Proceedings of the Workshop on Learning Software Organizations (LSO'99), pp. 45–55, Kaiserslautern, Germany, June 1999.

Goldmann, S., Kötting, B.: Software Engineering over the Internet, IEEE Internet Computing, Vol. 3, No 4, July/August 1999, pp 93 – 94, IEEE Computer Society Press, 1999

Goldmann, S., Kötting, B.: Working Group Report of the 2nd Workshop on Coordinating Distributed Software Development Projects, Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '99), IEEE Computer Society Press

Goldmann, S., Münch, J., Holz, H.: MILOS: A Model of Interleaved Planning, Scheduling, and Enactment, ICSE 99 Workshop on Software Engineering over the Internet, 1999

Goldmann, S., Münch, J., Holz, H.: Distributed Process Planning Support with MILOS, Proceedings of the 11th International Software & Engineering and Knowledge Engineering Conference (SEKE '99), Knowledge Systems Institute, ISBN 1-891706-01-2, 1999

Goldmann, S., Münch, J., Holz, H.: A Meta-Model for Distributed Software Development, Proceedings of the 8th IEEE International Workshops on Enabling

Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’99), IEEE Computer Society Press, 1999

Hoang, B., Maurer, F.: A Distributed Web Based Software Configuration Management System, Proceedings of the ICSE ‘99, Los Angeles, California

Kötting, B., Maurer, F.: A Concept for Supporting the Formation of Virtual Corporations through Negotiation, Proceedings of the WET ICE '99, IEEE Computer Society Press

Kötting, B., Maurer, F.: Approaching Software Support for Virtual Software Corporations, Proceedings of the ICSE ‘99, Los Angeles, California

Kötting, B., Maurer, F.: A Concept for Supporting the Formation of Virtual Corporations through Negotiation, Proceedings of the WET ICE '99, IEEE Computer Society Press.

Kötting, B., Richter, M. M., Goldmann, S.: Flexibles Workflowment in Software-Engineering Prozessen, Forschungsforum Wissensmanagement und schnelle Produktentwicklung, Stuttgart 1999

Maurer, F., Dellen, B., Holz, H.: Process Support for Virtual Software Organizations, Proceedings of the 11th International Software & Engineering and Knowledge Engineering Conference (SEKE ‘99), Knowledge Systems Institute, ISBN 1-891706-01-2, 1999

Maurer, F., Holz, H.: Process-Centered Knowledge Organization for Software Engineering, Aha, D.W., & Munoz Avila, H. (Eds.): Exploring Synergies of Knowledge Management and Case-Based Reasoning: A 1999 AAAI Workshop (Working Notes). (Technical Report AIC-99-008). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence

Maurer, F., Holz, H.: Process-Oriented Knowledge Management For Learning Software Organizations, Proceedings of the 12th Knowledge Acquisition Workshop (KAW '99), Banff, Canada

Maurer, F., Succi, G., Holz, H., Kötting, B.Goldmann, S., Dellen, B.: Software Process Support over the Internet, Proceedings of the 21 th International Conference on Software Engineering (ICSE ’99), ACM Press, ISBN 1-58113-074-0, pp. 642-645,

F. Maurer, H. Holz: Process-oriented Knowledge Management for Learning Software Organizations, Proceedings KAW-99, Banff, Canada, 1999.

F. Maurer, H. Holz: Knowledge Management for Learning Software Organizations, Proceedings Knowtech Forum 1999, Potsdam, Germany, 1999.

F. Maurer, G. Succi, H. Holz, B. Kötting, S. Goldmann, B. Dellen: Software Process Support over the Internet, Proceedings International Conference on Software Engineering (ICSE 99), IEEE press, 1999.

M. M. Richter, K. D. Althoff: Similarity and Utility. Mathematische Methoden der Wirtschaftswissenschaften, (eds. Wolfgang Gaul, Martin Schader), ISBN 3-7908-1208 Physica-Verlag, Heidelberg, 1999.

2000 (Distributed MILOS-System, Extensions)

A. Abecker, S. Decker, F. Maurer Organizational Memory and Knowledge Management - Guest Editorial, Journal on Information Systems Frontiers (ISF), Volume 2, Issue 3/4, October 2000.

Goldmann, S., Münch, J., Holz, H.: Distributed Process Planning Support with MILOS. Int. Journal of Software Engineering and Knowledge Engineering, October 2000

B. Kötting, F. Maurer: Building a virtual marketplace for software development tasks, Proceedings of the ICSE-2000nWorkshop "Software Engineering over the Internet",

B. Kötting. M. Schaaf: Merging Project Planning and Web-enabled Dynamic Workflow for Software Development. IEEE Internet Computing, Special Issue on Internet Based Workflow, June 2000.

F. Maurer, B. Dellen, F. Bendeck, S. Goldmann, H. Holz, B. Kötting, M. Schaaf: Merging Project Planning and Web-Enabled Dynamic Workflow Technologies, IEEE Internet Computing, May/June 2000.

F. Maurer, W. Regli: Knowledge networking - Weaving the Web of Reason, IEEE Internet Computing, Sep/Oct 2000.

P. Nour, H. Holz & F. Maurer: Ontology-based Retrieval of Software Process Experiences, Proc. of the ICSE-2000 Workshop "Software Engineering over the Internet", 2000.

2001 (Adding concepts of process based knowledge management, Java applications, extreme programming):

F. Bendeck, B. Kötting, M. Schaaf, F. Maurer, M. Valenti, M. Robert: Engineering of e-Business Applications & Infrastructure and Applications for the Mobile Internet, Proceedings WETICE 2001, IEEE, 2001.

H. Holz, A. Könnecker and F. Maurer. Task-Specific Knowledge Management in a Process-Centred SEE, in K.-D. Althoff, R.L. Feldmann and W. Müller (eds.): Advances in Learning Software Organizations, Proc. of the Third International Workshop (LSO 2001), Kaiserslautern, Germany, September 12-13, 2001. Springer, LNCS 2176, ISBN: 3-540-42574-8.

B. Kötting, F. Maurer: Approaching Support for Internet-based Negotiation on Software Projects, Proceedings WETICE 2001, IEEE, 2001.

B. Kötting, F. Maurer: Approaching negotiation automation for software development companies, Proceedings of the ICSE-2000 Workshop "Software Engineering over the Internet", 2001.

F. Maurer: Engineering Web Applications with Java, In: John J Marciniak (Ed.): Software Engineering Encyclopedia, Wiley, December 2001.

Frank Maurer, Carmen Zannier: 4th ICSE Workshop on "Software Engineering over the Internet", ACM Software Engineering Notes, Vol. 26 No. 6, Nov 2001 (p. 29-30)

F. Maurer: Process-Centered Knowledge Management and Case-Based Reasoning. Proc. of the Workshop on "Case-Based Reasoning Approaches for Process-Oriented Knowledge Management" held at the International Conference on Case-based Reasoning (ICCBR 2001), Vancouver, BC 2001.

F. Maurer, S. Martel: Supporting Distributed Extreme Programming (Extended Abstract), OOPSLA 2001 Companion, Tampa, Florida, 2001.

F. Maurer, J. Wittevrongel: Providing Software Engineering Services for Virtual Software Organizations (Work in Progress), Proceedings of the ICSE-2000 Workshop "Software Engineering over the Internet", 2001.

M. Schaaf, F. Maurer: Integrating Java and CORBA: A Programmer's Perspective, IEEE Internet Computing, Vol. 5, No. 1, January/February 2001.

M. Schaaf, F. Bendeck, Ph. Nour: Using MILOS for Dependency Management in UML-based SE-Processes, Proc. Object-Oriented Information Systems

J. Wittevrongel, F. Maurer: Using UML to Partially Automate Generation of Scenario-Based Test Drivers, Proceedings Object-Oriented Information Systems Applications, Proceedings WETICE 2001, IEEE, 2001.

J. Zettel, F. Maurer, J. Münch, L. Wong: LIPE: A Lightweight Process for E-Business Startup Companies based on Extreme Programming, Proceedings PROFES 2001, Springer, 2001.

2002 (Peer-to-peer integration; Connecting MILOS and a CBR-system; adding further concepts for planning support)

S. Bowen, F. Maurer: Designing a Distributed Software Development Support System Using a Peer-to-Peer Architecture. Proc. of the Workshop on Cooperative Supports for Distributed Software Engineering Processes held at the 26th IEEE Annual International Computer Software and Application Conference 2002 .

S. Bowen, F. Maurer: Process Support and Knowledge Management for Virtual Teams Doing Agile Software Development, Proceedings of the Workshop on Cooperative Supports for Distributed Software Engineering Processes held at the 26th IEEE Annual International Computer Software and Application Conference 2002 (3 pages).

S. Bowen, F. Maurer: Using Peer-to-Peer Technology to Support Global Software Development – Some Initial Thoughts. ICSE 2002 Workshop on Global Software Development, http://www.cis.ohio-state.edu/~nsridhar/ICSE02/GSD/, 2002 .

Frank Maurer, Harald Holz: Integrating Process Support and Knowledge Management for Virtual Software Development Teams, Annals of Software Engineering, Special Volume on Process-Based Software Engineering, Kluwer Academic Publishers, Vol 14, 2002

F. Maurer: Supporting Distributed Extreme Programming, Proceedings Agile Universe/ XP Universe 2002, Springer, 2002 .

G. Melnik, F. Maurer: Perceptions of gile Practices: A Student Survey, Proceedings Agile Universe/ XP Universe 2002, Springer, 2002 .

F. Maurer, S. Martel: Agile Software Development: An Industrial Case Study, ICSE 2002 Workshop on Iterative, Adaptive, and Agile Processes .

F. Maurer. S. Martel: Process support for distributed extreme programming teams, ICSE 2002 Workshop on Global Software Development, http://www.cis.ohio-state.edu/~nsridhar/ICSE02/GSD/, 2002 .

Frank Maurer, Sebastien Martel: Extreme Programming- Rapid Development for Web-Based Applications, IEEE Internet Comp. Magazine, Jan/Feb 2002 (pp. 86-90).

S: Bowen, F. Maurer: Using Peer-to-Peer Technology to Support Global Software Development – Some Initial Thoughts, ICSE 2002 Workshop on Global Software Development, http://www.cis.ohio-state.edu/~nsridhar/ICSE02/GSD/, 2002 .

2003 (Agile software processes, Scaling, Lightweight knowledge management)

F. Maurer, D. Wells (eds.): Proceedings XP Agile Universe 2003, Lecture Notes in Computer Science 2753, Springer, 2003.

K. Read, F. Maurer: Issues in Scaling Agile Using an Architecture-Centric Approach: A Tool-Based Solution, Proceedings XP Agile Universe 2003, Springer 2003, p. 157-165.

G. Melnik, F. Maurer: Introducing Agile Methods in Learning Environments: Lessons Learned, Proceedings XP Agile Universe 2003, Springer 2003, p. 187-198.

T. Chau, F. Maurer, G. Melnik: Knowledge Sharing: Agile Methods vs. Tayloristic Methods, Proc. WETICE 2003, IEEE 2003.

F. Paetsch, A. Eberlein, F. Maurer: Requirements Engineering and Agile Software Development, Proc. WETICE 2003, IEEE 2003.

C. Zannier, F. Maurer: Tool Support for Complex Refactoring to Design Patterns, Proc. XP 2003, Springer 2003.

Frauke Paetsch: Requirements Engineering in Agile Software Development, Diploma Thesis, FH Mannheim (March 2003)

H. Holz, F. Maurer: Knowledge Management Support for Distributed Agile Software Processes, Invited contribution for the Proceedings of the Workshop on Learning Software Organization 2002, Springer (to be published).

D. J. Reifer, F. Maurer, H. Erdogmus: Scaling Agile Methods, IEEE Software, July/August 2003 (Vol. 20, No. 4), p. 14-16.

K. Read, F. Maurer: Developing Mobile Wireless Applications, IEEE Internet Computing, January/February 2003 (Vol. 7, No. 1), p. 81-86.