

Management of Development Processes – An Evolutionary Approach

*Ansgar Schleicher, Diss. RWTH Aachen
Deutscher Universitäts-Verlag, Wiesbaden, 2002*

The management and coordination of development processes in technical domains is a difficult task. Development teams are growing in size and the advancing globalization imposes a higher pressure to develop more competitive products faster and cheaper. The employment of process management systems (e.g. workflow management or project management systems) promises improvements in terms of planning and monitoring support or even developer guidance and coordination.

However, long-term studies have revealed that current process support technology, which has proven useful for routine, repetitive processes, is insufficient for development process management. Development processes are of an inherently dynamic nature. Changing requirements and standards, unexpected feedback or the dependence of the process on its own intermediate results disable the a-priori planning of the complete process and lead to its continuous evolution. Furthermore, the knowledge about development processes is often heterogenous with respect to its preciseness and completeness. Available process management systems are incapable of addressing the inherent evolution and the heterogeneous available process knowledge. Both, managers and developers agree on their inflexibility, which disables managers to react to ad-hoc situations and hinders developers in their creativity.

This book describes the concepts behind and the implementation of a process management system addressing these issues of supporting process evolution and the handling of heterogenous process knowledge. The concepts are based on our studies of real development processes in the domain of chemical engineering, which have led to new requirements. These requirements and an ideal system structure are described within a conceptual framework, which identifies and interrelates the necessary concepts and builds the terminological and conceptual foundation for this book. The framework proposes a four-layer view onto process management. The first layer reflects the real process being performed by the developers. The real process is mapped and guided within the process model instance layer. Reusable process knowledge is kept within the process model definition layer. Process model definitions can be instantiated to create new process model instances. The syntax and semantics of the process model definition and instance are defined within the process meta model layer, where an enactable process modeling language is offered.

The essential contributions of this framework and its realization with respect to process evolution and process knowledge management support are as follows:

- Process model instances, which map and guide the real process, can be created, maintained and enacted in an interleaved fashion. A process manager is enabled to plan the process as far ahead as the current situation permits. He can complete and revise the plan as necessary, even when the process is already performed by the developers.
- The framework does not enforce the consistency of process model definitions and instances. As a result, a process manager or developer may deviate from the plan, if necessary. The system detects occurred inconsistencies and signals and explains them to the process participants. Inconsistencies may remain persistent, but can also be removed by applying the interleaved manipulation and enactment facilities.
- Process model instances can be created from process model definitions with any degree of preciseness. By consequence, the full range of process model instances, from

ad-hoc to consistent with respect to a given process model definition, can be supported. This enables the discovery and the improvement of a process (model) through the application of the process management system.

- Process model definitions can be inferred from ad-hoc or inconsistent process model instances. These implicitly carry new knowledge about a process, which has not been modeled before. An inference tool aids the process modeler in discovering new knowledge about a process and in creating new or revising existing process model definitions.
- The process model definition is expressed and maintained in the Unified Modeling Language, which enables abstract, visual and object-oriented modeling of processes. Changes to the process model definition, which may be induced by the inference mechanism or a process modeler, can be applied on the level of separate packages through a fine-grained versioning mechanism.
- Changed process model definitions may be propagated onto their enacting instances. A migration technique allows for the migration of a process model instance from an old to a new version. The proposed migration algorithm is guaranteed to be applicable, because it is based on the inconsistency toleration mechanism. We have particularly considered the requirement of leaving the manager in control of his managed process and of supporting economically and motivationally sensible migration of instances.

As this description suggests, the singular concepts are highly interrelated. Their integration yields synergetic effects. These are unobtainable, when only realizing a cutout of the proposed framework. The discovery of new process knowledge, the interleaved editing and enactment of process model instances within or outside of the constraints imposed by the process model definition, the latter's continuous and fine-grained changeability and the guaranteed migratability of process model instances amount to a process management system offering roundtrip process evolution support and wide spectrum process knowledge management capabilities.

Referee: Prof. Dr.-Ing. Manfred Nagl, RWTH Aachen
Coreferee: Prof. Dr. Theo Härder, Universität Kaiserslautern
Oral exam: March 8, 2002