Process Modeling Principles

Marie-Pierre Gervais & …

UPMC-LIP6 & …

Context of this work

- The present courseware has been elaborated in the context of the MODELPLEX European IST FP6 project (http://www.modelplex.org/).
- Co-funded by the European Commission, the MODELPLEX project involves 21 partners from 8 different countries.
- MODELPLEX aims at defining and developing a coherent infrastructure specifically for the application of MDE to the development and subsequent management of complex systems within a variety of industrial domains.
- To achieve the goal of large-scale adoption of MDE, MODELPLEX promotes the idea of a collaborative development of courseware dedicated to this domain.
- The MDE courseware provided here with the status of open-source software is produced under the EPL 1.0 license.

Outline

- Basic concepts of process modeling
- Comparison of process technology domains
- Survey of UML-Based software process modeling languages

I - Basic concepts of process modeling [Bendraou07a]

- Process
- Process Model
- Process Modeling Language

Process: a general definition

- ISO 9000 Quality Management, 1998 [ISO98]
  - "A process uses resources to transform inputs into outputs. In every case, inputs are turned into outputs because some kind of work, activity or function is carried out"
- The term "process" is used in various domains (and by various communities in each domain)
  - Definitions may differ slightly or may be completely different from one domain into another
    - e.g. Computer Science domain Vs. Chemical Industry domain
- We focus on three communities of the computer and information science domain
  - **Workflow management** community
  - **Business Process Management** community
  - **Software Process Engineering** community

Process definition in Workflow management community

- Workflow Management Coalition, Terminology and Glossary, WFMC-TC-1011, 1999 [WFMC99]
  - "a **process** is a formalized view of a **business process**, represented as a coordinated (parallel and/or serial) set of process activities that are connected in order to achieve a common goal"
  - "a **business process** is a set of one or more linked procedures or activities, which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships"

Process definition in Business process management community

- Many definitions, e.g. [Bastida05], [Davenport93], [Johansson93], [OMG06]

- [Davenport93] : most representative definition (in our view)
  - "a business process is a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what. A **process** is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for process actions"

Process definition in Software Process Engineering community

- Many definitions, e.g. [Lonchamp93], [Humphrey89], [Sommerville07], [Fuggetta00], [OMG05], [ISO06]

- [Lonchamp93] : most representative definition
  - "a software process is the set of partially ordered process steps, with sets of related artifacts, human and computerized resources, organizational structures and constraints, intended to produce and maintain the requested software deliverables"

- [Humphrey89] : most cited in literature
  - "a software engineering process is the total set of software engineering activities needed to transform a user's requirements into software"

Process definition: Summary

- Similarities between the three communities
  - Process: (roughly) a set of activities required to transform inputs into outputs
- Differences between the three communities
  - The objectives the process tends to attain
    - e.g. a reliable product or service, an added value in terms of benefits or customer satisfaction, a rigorous agent coordination, etc.
  - The means and vocabulary are proper to each process domain
  - The result of applying the process

Process Models

- [Curtis92]
  - "a process model is an abstract description of an actual or proposed process that represents selected process elements that are considered important to the purpose of the model and can be enacted by a human or a machine"
- [Scacchi01]
  - A software process model represents a networked sequence of activities, objects, transformations and events that embody strategies for accomplishing software evolution
- Constituents of software process models
  - Activity, artifact, role, human, tool

Process Models - Purposes

- Facilitate human understanding and communication
- Facilitate process reuse and repeatability
- Support of process analysis and improvement
- Support process management (process's status)
- A basis automate agents coordination and guidance
- A basis to automate execution support of the
- process (automation of repetitive and non interactive steps)

Process Models - views (1)

- One view cannot capture all process concerns!
- The **activity** view: focuses on the types, structure and properties of the activities in the process and their dependencies.
  - Use: Process understanding and reasoning, scheduling purposes and monitoring, possibly a basis for process automation, etc.;
- The **product** view: describes the types, structure and properties of process's artifacts.
  - Use: identification of process's deliverables, relations and dependencies between process's artifacts, the impact of an artifact changes on the other artifacts, etc.;

Process Models - views (2)

- The **resource** view: describes the resources (human or physical) either needed from or supplied to the process, which is relevant from a managerial perspective;
  - Use: project planning, resource allocations, etc.
- The **role** view: describes a particular set of resources, namely skills that performers supply and the responsibilities they accept.
  - Use: Process planning and management, access rights affectations, etc..

Process Modeling Languages

- [Zamli01]
    - "a process modeling language is a language used to express process models"
- Process modeling language (PML) requirements
    - From the consideration: The PML is a modeling language
        - Expressiveness (semantic richness), understandability, modularity
        - [Conradi95], [Kellner89]
    - From the consideration: The PML is a programming language
        - Executability, formality
        - [Osterweil87]

II - Comparison of process technology domains [Bendraou07b]

Process definition and goal of modeling processes

Process composition

Domain and process characteristics

Process model elements and domain’s context and scope

Relationship with other domains

Process definition and goal of modeling processes

		Software Process Engineering (SPE)	Business Process Management (BPM)	Workflow Management (WfM)
Definition of the term Process		The set of partially ordered process steps, with sets of related artifacts, human and computerized resources, organizational structures and constraints, intended to produce and maintain the requested software deliverables [Lonchamp93]	A specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs. Processes are the structure by which an organization does what is necessary to produce value for its customers [Davenport93]	A formalized view of a business process, represented as a coordinated (parallel and/or serial) set of process activities that are connected in order to achieve a common goal [WFMC99]
Goals of Modeling Process	Primary Goal	To facilitate and to support the development of high-quality software more quickly and at lower cost	Creation/addition of Value for the customer and for the organisation	Automatic routing of artifacts across process activities and participants
	Secondary Goals	Process planning, understanding, analysing, measuring, configuring, reusing, executing and improving	Support for human understanding, communication, process improvement, analysis, simulation and execution	Saving of time by automatic task and artifacts affectations, detection of process bottlenecks

Process composition

	Software Process Engineering (SPE)	Business Process Management (BPM)	Workflow Management (WFM)
Process Composition (Sub-Processes or Phases)	The Software Production Process: represents the process being actually performed by software developers and tools The Meta-Process: consists of the activities of modeling the process, managing the process, support for its execution and improvement	The Core Process: the primary activities that converted raw materials into finished products and sold and delivered them The Support Process: the activities of process automation, managing resource and process, analysis and improvement	There is no firm consensus in the literature on process phases in the Workflow domain. The WfMC define two phases: Process Definition Phase: designates the time period when manual and/or automated (workflow) descriptions of a process are defined and/or modified electronically Process Execution Phase: the time period during which the process is operational, with process instances being created and managed

Domain and process characteristics

	Software Process Engineering (SPE)	Business Process Management (BPM)	Workflow Management (WFM)
Domain and Process Characteristics	- complex - exception driven - highly unpredictable since they depend too much on too many people and circumstances - Not all activities are supported by automated tools - They depend on communication, coordination and cooperation within a predefined framework - Their success depends on user involvement and the coordination of many roles - They may take a long time and are subject to changes during this time	- may cross functional/ organisational boundaries - High degree of natural concurrency—many interrelated tasks are running at any given point of the business process - There is a requirement to monitor and manage the overall business process - complex, highly dynamic and unpredictable - often long running - need user interactions - may need to migrate in the event of hardware failure or for performance - stateful - they have customers (internal or external)	- Workflow systems are suitable for supporting rigidly structured, well-defined and repeatable business processes - Processes are more often implemented in terms of proprietary code and procedures - Business rules are hard-coded. Once the process execution is lunched, it is not possible to modify them - Workflow systems are tightly coupled through customized APIs with software and applications used during the process - Workflows are limited in modeling all aspects of business processes - Not all workflow systems provide graphical representations

Process model elements and domain's context and scope

		Software Process Engineering (SPE)	Business Process Management (BPM)	Workflow Management (WFM)
Process Model Elmt	General Process Element	Activity, Artifact, Role, Tool and Agent	Activity, Artifact, Role, Tool and Agent	Activity, Artifact, Role, Tool and Agent
	Domain-specific Process Element	Deliverable, Constraint, Milestone, Guidance, Team, Phase, Iteration, Lifecycle, etc.	Organizational Unit or Entity, Customer, Software, Goal, Process Owner, Service, Business Rule, Event, Message, Condition, Date, Transaction, Sub-Process, Time, etc.	Work Item, Work List, Task (automated activity), Instance (of process or activity), Deadline, Procedure, Rule, Application, Event, etc.
Domain's context and scope		Organisations having as main business the development and maintenance of software. In addition, organisations that tend to retain their expertise and strategic business by developing their own software. The scope of the process is the organisation.	Any organisation, what ever its business, that wants to create/add value to its customers. Business processes span many organization's units as well as many other organizations	Workflows are often Organisation/Department specific

Relationship with other domains

	Software Process Engineering (SPE)	Business Process Management (BPM)	Workflow Management (WFM)
Relationship with other domains	- Software Processes are special cases of Business Processes. - Software Processes use the Workflow technology to automate the routing of artifacts, the coordination of process activities and roles	- A Business process can be a Software Process since the BPM domain is richer than the SPE. - BPM is a superset of the Workflow technology. That latter is used to automate process's repeatable tasks and for artifacts routing	- The Workflow Technology is used by BPM and by SPE to automate process's repeatable tasks and for routing artifacts.

III - Survey of UML-Based software process modeling languages

UML as a basis for Process Modeling Language

SPEM1.1

SPEM2.0

Framework specialization approaches

Two layers approaches

Combining meta-modeling and executability

UML as a Process Modeling Language (PML)?

- A Privileged candidate as PML
- Understandability
    - High-Level Constructs and Graphical Representations
    - Multiple perspectives/views
        - Diagrams suitable for process modeling: Class diagrams, Activity and State machine diagrams
- Expressiveness
    - Concepts and mechanisms to ensure Proactive & Reactive Control
        - Ensures more than 16/20 control flow patterns [Wohed05],
    - Advanced elements such as loops, conditional branches and exceptions
- Executability
    - Concepts with execution semantics (Activity & Action packages)
    - However no universally UML virtual machine adopted!
- Last but not least: UML is a Standard and a widely adopted formalism
    - Easier adoption
    - Leveraging existing tools and training supports

Survey on UML-based SPMLs – Evaluation Criteria

- Expressiveness (Semantic Richness)
  - Description of basic process elements
    - **Activity**, **Artifact**, **Role**, **Human** and **Tool**
  - *Activities and steps sequencing*
    - **Proactive control Vs. Reactive control**
- Understandability
  - Code like descriptions Vs graphical representations
  - Multiple perspectives
- Modularization
  - The ability to combine different chunks of processes

Survey on UML-based SPMLs – Evaluation Criteria

- Executability
  - Constructs with operational semantics
  - Support of process execution and simulation
- Flexibility
  - To be able to adapt and customize your Software Process to specific projects
    - Adding new kinds of process elements
      - E.g., roles, artifacts, activities, etc.
  - Extending the SPML at lower costs
    - No need to build a new interpreter/compiler for the SPML
  - Humans as deciders for the process workflow
    - Modification of the process model at runtime

SPMLs concerned by the survey

- Standards
  - SPEM1.1 [OMG05]
  - SPEM2.0 [OMG07a]
- Framework specialization approaches
  - Di Nitto's et al. Approach [Di Nitto02]
  - Promenade [Franch98]
- Two layers approaches
  - Chou's Approach [Chou02]
- Combining meta-modeling and executability
  - UML4SPM [Bendraou06]

Standards - SPEM1.1

- the OMG's standard for software process modeling
  - Adopted by early 2005
- Comes in form of a MOF 1.3-compliant metamodel that extends concepts from UML1.4 and an UML1.4 Profile
- The building block of the SPEM metamodel is the Process Structure package
  - defines the main structural elements from which a process description may be constructed

Standards - SPEM2.0

- The recently adopted revision of SPEM1.1 [OMG07a]

- SPEM2.0 comes with a new attractive vision
    - Separating all the aspects, contents and materials related to a software development methodology from their possible instantiation in a particular process

- SPEM2.0 comes in the form of MOF-compliant metamodel that reuses UML2.0 Infrastructure and UML2.0 Diagram Interchange specifications.
    - No concept from the UML2.0 Superstructure [OMG 07b] is reused.
        - Was specified as a mandatory requirement in the RFP!!!
    - The standard comes also in form of a UML Profile
        - The OCL (Object Constraint Language) rules were not provided with the profile.

Framework specialization approaches

- Principles : SPMLs are the result of the specialization of an Object-Oriented framework, with a modeling layer in UML and an implementation layer partially generated from the UML layer
- These approaches does neither provide a new metamodel nor extend the UML metamodel
  - The UML class diagram is used as a mean to represent process constituents in terms of UML classes
  - The Activity diagram is used to represent the flow of work
  - The state machine diagrams (if used) to represent product and activity states
- Main proposition: Di Nitto and Promenade approaches

Two layers approaches – Chou's approach

- Principle: UML diagrams are used for documenting and modeling the process. A programming language is used to represent these descriptions in a machine readable format
  - No automatic code generation is provided!

- Main proposition: the Chou's approach:
  - The UML class and activity diagrams are used for modeling the process
  - A proprietary programming language is used to support process execution

Combining meta-modeling and executability - UML4SPM Language

- UML4SPM comes in form of a MOF-compliant metamodel which extends a subset of UML2.0 (Activity and Action packages)
    - A metamodel, a notation and an execution semantics
- Provides the possibility to define executable and simulable software process models

Summary of the survey of current UML-Based SPMLs

Approaches Requirements		SPEM1.1	SPEM2.0	Di Nitto's et al. Approach	Promenade
Expressiveness					
Process Elements			Depends on the Compliance point used	Instances of the UML1.3 *Class* metaclass	Instances of the UML1.3 *Class* metaclass
Activity		Work Definition, Activity	Activity / Task Definition	Activity	Task
Role		Process Role	Role Use / Role Definition	No	Role
Artifact		WorkProduct	WorkProduct Uses / WorkProduct Definition	Artifact	Document
Agent		No	No	Human Agent	Agent
Tool		No	Tool Definition	Software Agent	Tool
Activities/ Actions Coordination	Proactive Control	*start-start*, *finish-start* or *finish-finish*. The *start-finish* is lacking	*start-start*, *finish-start* or *finish-finish* and *start-finish*	*finish-start* and *start-start*. *finish-finish* and *start-finish* are lacking	*start-start*, *finish-start* or *finish-finish*. the *start-finish* is lacking
	Reactive Control	No	No	Use of events	No
Exception Handling		Not addressed	Not addressed	Not addressed	Not addressed
Understandability		Good.	Lack. Very complex. One has to deal 3 compliance points, 7 packages and 75 metaclasses.	Lack. No link between the diagrams.	Lack. Process modeler has to combine between class diagrams and precedence graph. No link between the two diagrams

Summary of the survey of current UML-Based SPMLs

Approaches Requirements	SPEM1.1	SPEM2.0	Di Nitto's et al. Approach	Promenade
Executability	Lack. Out of scope	Lack. Not addressed.	Lack. No details on how the code is generated and how different information from the different diagrams are integrated to the Java code. If the state diagram is not complete, code has to be added manually after generation	Lack. Not addressed
Modularization	Lack. A *Unification* phase is needed (i.e., renaming of process elements)	Good. Use of *Variability* mechanisms.	Not addressed	Lack. Based on renaming classes from different class diagrams
Metamodel/ Profile	A MOF-Compliant metamodel extending UML 1.4 concepts / A UML1.4 Profile is proposed	A MOF-Compliant metamodel extending the UML2.0 Infrastructure / A UML2.0 Profile Lack. OCL constraints are not provided	Lack. No metamodel, no profile provided. Authors provide a predefined UML1.3 class diagram	Lack. No metamodel, no profile provided. Authors provide a predefined UML1.3 class diagram

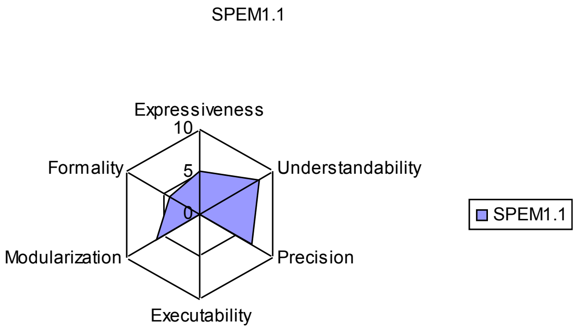
Summary of the survey of current UML-Based SPMLs

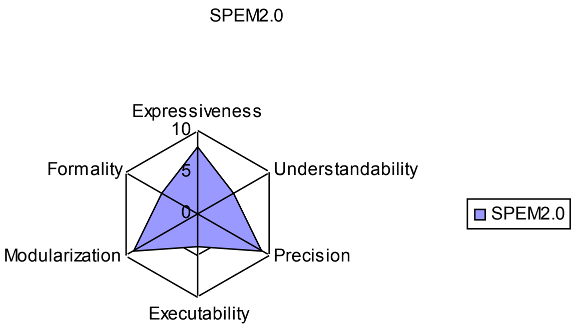
Approaches	UML4SPM
Requirements	
Semantic Richness	
Process Elements	
Activity	Software Activity. Many used-defined kinds can be added thanks to "kind" metaclass
Role	Responsible Role
Artifact	WorkProduct
Agent	Agent and Team
Tool	Tool
Activities/Actions Coordination	
Proactive Control	Ensured thanks to the combination of control flow, object flow, Invocation Actions and control nodes. Kinds of precedence ensured: start-start, finish-start or finish-finish and start-finish.
Reactive Control	Ensured through events (Message, Change and Time events) and actions (AcceptEvent and SendSignal actions)
Exception Handling	Use of RaiseExceptionAction and ExceptionHandler concepts
Advanced Constructs (Tool Invocations, Agent Communication, Loops, Conditionals)	Loops, Conditional, Decision Nodes, DataStoreNode (WorkProduct persistency and versioning)

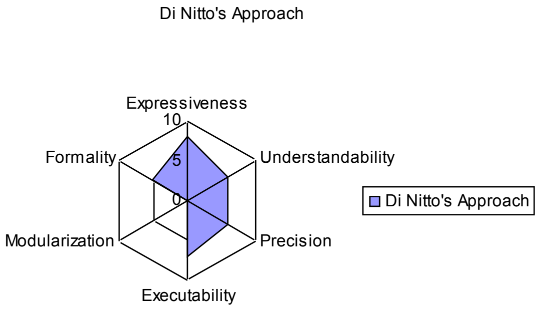
Summary of the survey of Current UML-Based SPMLs

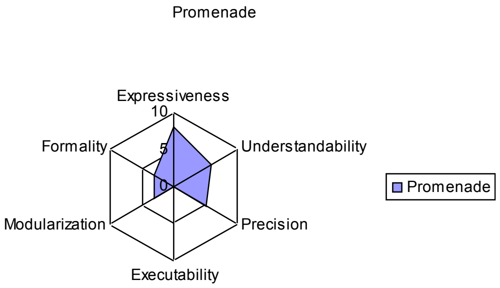
Approaches Requirements	UML4SPM
Understandability	Good: Simple metamodel with a minimal set of concepts, reuse of UML2.0 diagrams with a minor customization in order to represent features proper to software process modeling. Requires a minimum knowledge of UML2.0
Precision	Good: Two levels of abstractions. Process view for a global view of process's activities and Activity view for a detailed description of the activity in terms of actions. The unit of work is the Software Activity (can be specialized using the "Kind" metaclass i.e., a process, a phase, iteration, etc.). Action is the atomic unit of work in UML4SPM.
Executability	Good: Two approaches proposed. The first, transformation of UML4SPM process models towards BPEL for execution. The second, direct process model execution using UML4SPM process engine. Lack: Need for configuration phase in the BPEL approach. Lack of resource management support with the current UML4SPM process engine
Modularization	Through the use of CallBehaviorAction in order to compose between different activities. More configuration in case of complete process models.
Formality	Translational semantics to Petri-nets approach possible
Metamodel/ Profile	A MOF-Compliant metamodel extending UML2.0 Superstructure concepts. No profile is proposed.

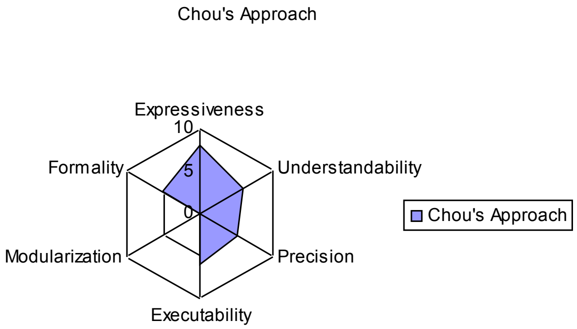
Summary

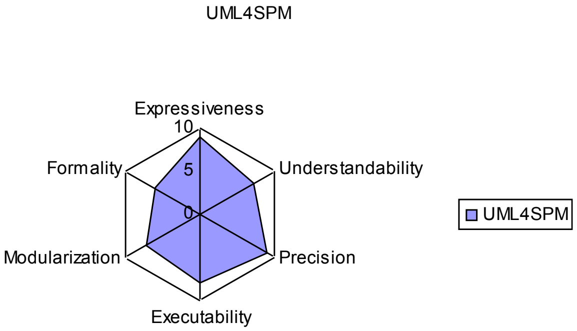

SPEM1.1
Expressiveness
10
5
0
Understandability
Precision
Executability
Modularization
Formality
□ SPEM1.1


SPEM2.0
Expressiveness
10
5
0
Understandability
Precision
Executability
Modularization
Formality
□ SPEM2.0


Di Nitto's Approach
Expressiveness
10
5
0
Understandability
Precision
Executability
Modularization
Formality
□ Di Nitto's Approach


Promenade
Expressiveness
10
5
0
Understandability
Precision
Executability
Modularization
Formality
□ Promenade


Chou's Approach
Expressiveness
10
5
0
Understandability
Precision
Executability
Modularization
Formality
□ Chou's Approach


UML4SPM
Expressiveness
10
5
0
Understandability
Precision
Executability
Modularization
Formality
□ UML4SPM

References (1)

- [Bastida05], L.Bastida Merino and G. Benguria Elguezabal1, "Business Process Definition Languages Versus Traditional Methods Towards Interoperability", Book chapter, COTS-Based Software Systems, Lecture Notes in Computer ScienceVolume 3412/2005, pages 25-35, ISBN: 978-3-540-24548-3, January 2005
- [Bendraou06], Bendraou R., Gervais M.P. and Blanc X, "UML4SPM: An Executable Software Process Modelling Language Providing High-Level Abstractions", in Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06), pp. 297-306, Hong Kong, China, , 2006
- [Bendraou07a], UML4SPM : un langagde de modélisation de procédés de développement logiciel exécutable et orienté modèle, PhD Thesis of University Paris 6, 6 septembre 2007 (in english)
- [Bendraou07b], Bendraou R., Gervais M.P. "A Framework for Classifying and Comparing Process Technology Domains", Proceedings of International Conference on Software Engineering Advances (ICSEA'07). IEEE Computer Society Press 2007
- [Chou02], S.-C. Chou, A process modeling language consisting of high level UML diagrams and low level process language, Journal of Object Technology 1, 2002, 4, pp. 137–163
- [Curtis92], W. Curtis, M. I. Kellner and J. Over. "Process Modelling". Communication of ACM, 35 (9), 1992, pp. 75-90
- [Davenport93], T. Davenport, Process Innovation: Reengineering work through information technology, Harvard Business School Press, 1993, Boston
- [Di Nitto02], Di Nitto E. et at. "Deriving executable process descriptions from UML", in Proc. of the 24th Inter. Conf. on Software Engineering (ICSE'02), Orlando, Florida 2002, ACM Press
- [Franch98], Franch, X., Ribó, J.M., "A Structured Approach to Software Process Modelling", Proceedings 24th EUROMICRO Conference, IEEE Computer Society Press, Los Alamitos Washington Brussels Tokyo (1998), 753-762
- [Fuggetta00], A. Fuggetta. "Software Process: A Roadmap". 22nd International Conference on Software Engineering (ICSE'2000), Future of Software Engineering Track, June 4–11, Limerick (Irlanda), ACM, 2000
- [Humphrey89], W.S. Humphrey. "The Software Engineering Process: Definition and Scope", in Proceedings of the 4th International Software Process Workshop on Representing and Enacting the Software Process, Devon, United Kingdom, 1989
- [ISO98], International Organization for Standardization. ISO/IEC, ISO 9000 Quality Management, 7th edition, 1998
- [ISO06], ISO SEMDM, "Software Engineering — Metamodel for Development Methodologies", ISO document, ISO/JTC 1/SC 7 ICS 35.080, May 2006
- [Johansson93], H.J. Johansson, et.al, Business Process Reengineering: BreakPoint Strategies for Market Dominance, 1993, John Wiley & Sons

References (2)

- [Lonchamp93], Lonchamp, J. 1993. A structured conceptual and terminological framework for software process engineering. In Proceedings of the 2nd International Conference on the Software Process (ICSP 2) (Berlin, Germany). IEEE Computer Society Press, Los Alamitos, Calif.
- [OMG05], OMG SPEM1.1, “Software Process Engineering Metamodel”, OMG document formal/05-01-06, January 2005, at http://www.omg.org.
- [OMG06], OMG BPMN, Business Process Modeling Notation final adopted specification, OMG document dtc/06-02-01, February 2006 at http://www.omg.org
- [OMG07a], OMG SPEM2.0, “Software Process Engineering Metamodel”, OMG document, final adopted specification, ptc/07-03-03, March 2007, at http://www.omg.org.
- [OMG07b], OMG UML, "Unified Modeling Language", Superstructure Specification, version 2.1.1., OMG document formal/07-02-04 , February 2007 at http://www.omg.org
- [Osterweil87], Osterweil L., "Software Processes Are Software Too" in Proc. of the 9th International Conference on Software Engineering (ICSE'9), New York, 1987, ACM Press
- [Scacchi01], W. Scacchi "Process Models in Software Engineering". In, J.J. Marciniak (ed.), Encyclopedia of Software Engineering, 2nd Edition, John Wiley and Sons, Inc, New York, Dec. 2001
- [Sommerville07], I.Sommerville, "Software Engineering 8", 2007, eighth edition, Addison-Wesley, ISBN 10:0-321-31379-8
- [Wohed05], Wohed, P., van der Aalst, W., Dumas, M., ter Hofstede,A. & Russell, N., "Pattern-based analysis of UML activity diagrams", in Proceedings of the 25th International Conference on Conceptual Modeling (ER’2005), Springer, Klagenfurt, Austria
- [WFMC99], WorkFlow Managment Coalition (WFMC),Terminology & Glossary, Document Number WFMC-TC-1011, Feb. 1999
- [Zamli01], K.Z. Zamli, , P.A. Lee, "Taxonomy of Process Modeling Languages". In: Proc. Of the ACS/IEEE International Conference on Computer Systems and Applications. IEEE Computer Society Press (June 2001) 435-437