Panel Introduction on
Software Process Metaprocess / Methodology

Reidar Conradi
Norwegian Institute of Technology, N-7034 Trondheim, Norway

Third European Workshop on Software Process Technology
(EWSPT’94)
Villard de Lans (Grenoble), France
7–9 February 1994

Phone: +47 73 593444, Fax: +47 73 594466, Email: conradi@idt.unit.no
Contents

1 Concepts for Software Process Modeling (PM) 3

2 Structure of the Meta-process: Process Lifecycle 5

3 Technologies for Process Evolution 6

4 This session 7
1 Concepts for Software Process Modeling (PM)

- External (software) process/application:
  a partially ordered network of interacting activities, aiming at producing a (software) product to fulfill certain goals.

  Three parts in a whole (software) process:

  - Production process: Produces items.
  - Meta-process: Maintains and evolves the whole process.
  - Process support (PML, process model and tools).

- Process element, at least five categories:
  - activities: simple/composite, I/O items, tools, and roles.
  - artifacts: persistent, constituting a product.
  - agents, humans described by roles.
  - tools.
  - projects (work contexts).
• **Production support**: formalisms, methods and tools e.g. to assist (software) production.

• **Production activities** work on production artifacts.

• **Meta-activities** work on “meta-artifacts”, thus changing:
  – the external production process (activities, production tools);
  – the external meta-process (human meta-roles, process tools);
  – the internal process model, i.e. *reflection*.

• **PML**: A process modeling language/formalism, covering both production processes and meta-processes.

• **Internal Process model**:
  a PML-description of process elements.
  Reflective? Submodels in sublanguages?

• **Process tools**: tools to acquire, define, analyse, refine, customise, instantiate, enact, monitor, and evolve process models.

• **Process support**: PML, process model(s), and process tools.
2 Structure of the Meta-process: Process Lifecycle

Coarse-level and fundamental meta-activities:

1. Technology Provision: PMLs, predefined models, tools, ...
4. Process Implementation, according to above specs:
   - Changing the process support:
     - Model:
       1. Make template process model (e.g. types).
       2. Make enactable model (e.g. activity network).
       3. Make enacting model (interpretable).
     - Tools: ...
   - Changing the real-world process:
     e.g. by migration tool.

No specific lifecycle model imposed: waterfall, evolutionary, ...

Allow incremental model creation (refinement, supplements) and evolution (customisation, correction, enhancements):
⇒ diffuse borderlines between model variations.
3 Technologies for Process Evolution

For the different process elements:
- What are the process parameters (What)?
- What reason, inputs, feedbacks etc. will bind them (Why)?
- How can the change be implemented and propagated (How)?
- By which meta-activity (When)?
- By which human roles (Who)?

Must cover the entire process lifecycle, (partly) independent of PML?

What mechanisms can be employed:
flexible architectures, reflection (interpretation with late binding), schema evolution and -versioning, encapsulation, ...

What methods can be employed:
normal software-engineering “repertoire” (rules, SADT, OO, ...), impact analysis, applying human roles (process managers), organisational aspects, assessment frameworks, ...

Meta-tool support for all this?
4 This session

Chair: Reidar Conradi, NTH.

Rapporteur: Geir Magne Høydalsvik, NTH.

Papers:

Jin Sa and Brian C. Warboys:
“Modelling Processes Using a Stepwise Refinement Technique” (ISPW7 example). 20 min.

Keith Phalp and M. Sheppard:
“A Pragmatic Approach to Process Modelling”. 10 min.

Tom Rodden et al.:
“Process Programming and Development Practise”.

Gerald Junkerman and Wilhelm Schäfer:
“A Design Methodology for Process Programming” (MERLIN).

Discussion: 60 min.