Software Process Research in Europe: A Brief Survey

Reidar Conradi
NTNU, Trondheim, Norway
conradi@idi.ntnu.no

FEAST’2000 Workshop, July 10-12, 2000,
Imperial College, London

Overview

1. Process Programming: PMLs and PSEEs
2. Software Quality and Metrics
3. Software process improvement (SPI)
4. Learning organizations
5. Related fields: enterprise modeling, BPR, CSCW, workflow, middleware
6. The way ahead: Some challenges
7. Six Theses on Software Process Research
   (Conradi, Fuggetta, Jaccheri: EWSPT’98)
1. Process Programming: PMLs/PSEEs

- **PML = Process Modeling Language.**
  -- Network-based, rule-based, prog.lang.-based, and hybrids.
  -- Used for description, analysis, enactment, improvement, …
  -- Must model: activities (tasks), artifacts, roles/humans, tools, …
- **PSEE = Process-centered Software Engineering Environment.**
  -- Based on a PML and a model repository.
  -- Tools: model editor, analyzer, interpreter (process engine),
    plus glueware to connect to other tools.

- Much PML/PSEE work in 90’s, inspired by Osterweil’s ICSE’87 paper on **Process Programming** – but be aware Lehman’s response. And “da-capo” at ICSE’97.

- European workshop series 1990-2000 (7 **EWSPTs**), **PROMOTER**
  EU project w/two books: (Finkelstein94) (Derniame 99).

1a. Process Programming: PMLs

- Examples of European **PML efforts:**
  ALF (Derniame:Nancy): rule-based
  **TEMPO/APEL** in Adele (Estublier;Grenoble): OO, triggers
  **Process Weaver** (Fernström:Cap/Grenoble): task network
  **SOCCA** (Engels:Leiden): OO
  **SPELL** in EPOS (Conradi:Trondheim): OO, meta-types, coop.
  **SLANG** in SPADE (Fuggetta:Milano): Petrinets, OO, modules
  **OIKOS** (Ambriola:Pisa): rule-based agents, distribution
  **MELMAC/Leu** (Gruhn:Dortmund): petrinets
  **MERLIN** (Schäfer:Dortmund/Paderborn): rule-based, statecharts
  **PML** in PWI (Warboys:Manchester): OO, meta-level, persistent
  **M-PVL** (Rombach:Kaiserslautern): task network, multiple views
  **FEAST** language: system dynamics

- Four **industrial** products: PWI, Leu, ADELE, Process Weaver (?)
  But little industrial/practical penetration.
1a. Process Programming: PMLs (cont’d)

- Most companies use own and informal “boxologies”, perhaps on the Web (cf. Clara Purper in Bremen), mostly for documentation -- as part of a Quality System.
  At most using IDEF0 or other simple PMLs.
- Very little process enactment, except in ADELE (for SCM).
- Now “classic” PML work in Nancy, Manchester and Grenoble.
- Attempts at using UML as a PML, but almost hopeless (Westfechtel at ESEC’99).
- But many new agent- and XML-based formalisms/tools for PMLs/PSEEs, see next page.

1b. Process Programming: PSEEs

- PSEEs: Related to PML development, each PSEE its own PML.
- Some PSEEs sprang out of SCM/SDE work: ADELE, EPOS etc. Others had an AI/agent root: ALF, MERLIN, OIKOS.
- However, the initial PSEEs were too “closed” and restrictive in their architecture.
- Now: federated systems, new netware/middleware and event-based messaging, agents, XML etc.
- Cf. the Process Instance Evolution project (PIE), gluing ProcessWeb (on PWI) and ADELE with reflexive middleware.
- Cf. Endeavor at UCI (Taylor), event-based messaging in Milano (Fuggetta/Cugola), agent work in Bologna (Ciancarini), CAGIS project in Trondheim (Conradi) etc. etc.
- Workflow Mgm’t Coalition: XML protocol for process engines
- See also CSCW later.
2. Software Quality and Metrics

- Large European community on software quality and metrics, cf. Fenton’s work, many conference series, PERFECT project, …
- Project management is closely related.
- For many organizations: the above is the real process work.
- But problem with a “hard” measurement-oriented focus: projects are incommensurable and not repeatable, baselines are fuzzy – as for social sciences.
- Positivist vs. constructionist view (C. Floyd, Berlin)?
- Need spectrum of empirical methods, cf. new field of empirical software engineering (Basili, Rombach).

- Good coupling to SPI -- although classic quality work is about control, but SPI is about cultural change and learning.

3. Software Process Improvement

- *Fundamental assumption:* “better” process → better product.
- TQM for manufacturing, although emphasizing cultural learning and pervasive, general quality focus.
- Assessment/certification approaches: ISO-9001 (European-lead) and TickIT (UK), Trillium (BT), CMM (SEI), and BOOTSTRAP (EU project). Often management-oriented and top-down.
- Experience-driven approaches: QIP/EF/GQM (Basili, Rombach), partly Norwegian SPIQ. Often developer-oriented and bottom-up.
- Many European initiatives: ESSI project (200? PIES), national initiatives for SMEs in Norway (SPIQ), Denmark, Sweden, Finland, Netherlands, etc.
- ESI in Bilbao: Studies (ESSI, SPICE), SPI, software reuse, dissemination etc.
4. Learning organizations

- Large general community, and partly AI-oriented.
- **Experience Factory** (NASA-SEL, Basili and Rombach).
- Much work on **case-based reasoning** (IESE on Kaiserslautern).
- Much work on **remote and continuing** education: Open University, remote universities, corporate universities, …
- The gained knowledge is often in an (Web) **experience base**.
- **Externalization (easy)**: 1. Try to collect experiences (**data**), 2. Refine these into models and combine these into recommendations of best practice (**passive information**).
- **Internalization (hard!)**: 3. Disseminate, 4. Inject improved models into new, socialized work practices (**operational and active knowledge**).

4. Learning organizations (cont’d)

- So much interest and effort, but really hard.
- Ex. In Norway’s SPIQ project: only 2 of 4 companies succeeded with experience bases (no advanced knowl.repr.).
- Some **success factors** (FEAST position paper):
  1. Get cultural changes in becoming a learning organization.
  2. Organizational/champion stability: 3-5 years.
  3. Contributing to real business value: cost/benefit model?
  4. Incremental approach: avoid “big-bang” technology push.
- Risk of having an **information dump / graveyard**.
  Seek lean, relevant and updated information!
- Developers resist being told how to do their job; must demonstrate concrete usefulness on a daily basis --> **revolutionary learning**!
  Also solicit comments and feedbacks --> **two-way communication** (in CSCW-style), not only one-way flow.
- Try **Web technologies**, but how to maintain the contents and manage general information overload – via portals?
5. Related fields: enterprise modeling, BPR, workflow, CSCW, middleware

- **Enterprise modeling**: model “as-is” and “to-be” processes for engineering or general businesses, mostly for understanding, using simple PMLs. Done in European engineering companies.
- **Business Process Reengineering (BPR)**: a bit like above, but emphasis on big-bang changes.
- **Workflow**: “SPT” work by information systems community. Ex. Gustavo Alonso in Zurich, IBM’s FlowMark in Germany.
- **CSCW (groupware)**: Multidisciplinary community, emphasis on synchronous work against shared workspaces and with triggers, ethnographical studies. Ex. Lancaster group, Univ. Milano (Paoli), and many others.
- **Middleware**: enormous development in packaging technologies (COM, CORBA), Java/Jini/JavaSpaces, mobile agents/aglets, XML, messaging, federated and component-based systems, WAP. Ex. Work in PIE project, at Politecnico di Milano, …

6. The way ahead: Some challenges

**General advice:**
- Must couple academia and industry -- for the Nth time.
- And search for empirical validation (Tichy, Zelkowitz).
- Stop making useless SPT (next page).
- SPT must connect to other communities: CSCW, SPI, quality/metrics, social sciences.

**Research agenda:**
- Try to make a simple, graphical, common PML for companies?
- How different are software processes from other business processes: -- human-oriented, evolution/flexibility, complexity, …?
- Must give active help/guidance for decisioning and for negotiation.
- How to couple evolution, feedbacks, reflexivity, and exper. bases?
- How to cope with almost continuous changes – cf. Internet time?
- How to scale down SPT/SPI for SMEs?
- PMLs/PSEEs: Use XML/agents and new middleware for mobile and distributed work in federated systems (takes already place).
7. Six Theses on Software Process Research  
(Conradi, Fuggetta, Jaccheri: EWSPT'98)

- The problems and issues addressed by software process technology and workflow management are the same.
- Configuration management tools are the real “PSEEs”.
- The significant factor that distinguishes different classes of technologies is not the process domain in which they are supposed to be applied (e.g. information systems, software dev. etc).
- We have often tried to model what can’t be modeled, or what is not worthwhile and useful to model.
- SPI has the same nature as any other change/improvement initiative, and should apply results from org./behavioral sciences.
- So far we have substantially failed to demonstrate that SPT is useful. Also organizations still hesitate to adopt SPI methods.