
Discovering a pattern language for embedded machine control systems using architecture evaluation methods

Marko Leppänen (Tampere University of Technology),
Tommi Mikkonen (TUT), Johannes Koskinen (TUT)

Presentation outline, 10 minutes

- Background
 - Problems of EMCS design
 - Method for improving the designs, patterns
 - Collecting the pattern drafts
 - Pattern Workshops
 - Language Workshops
 - Conclusions
-

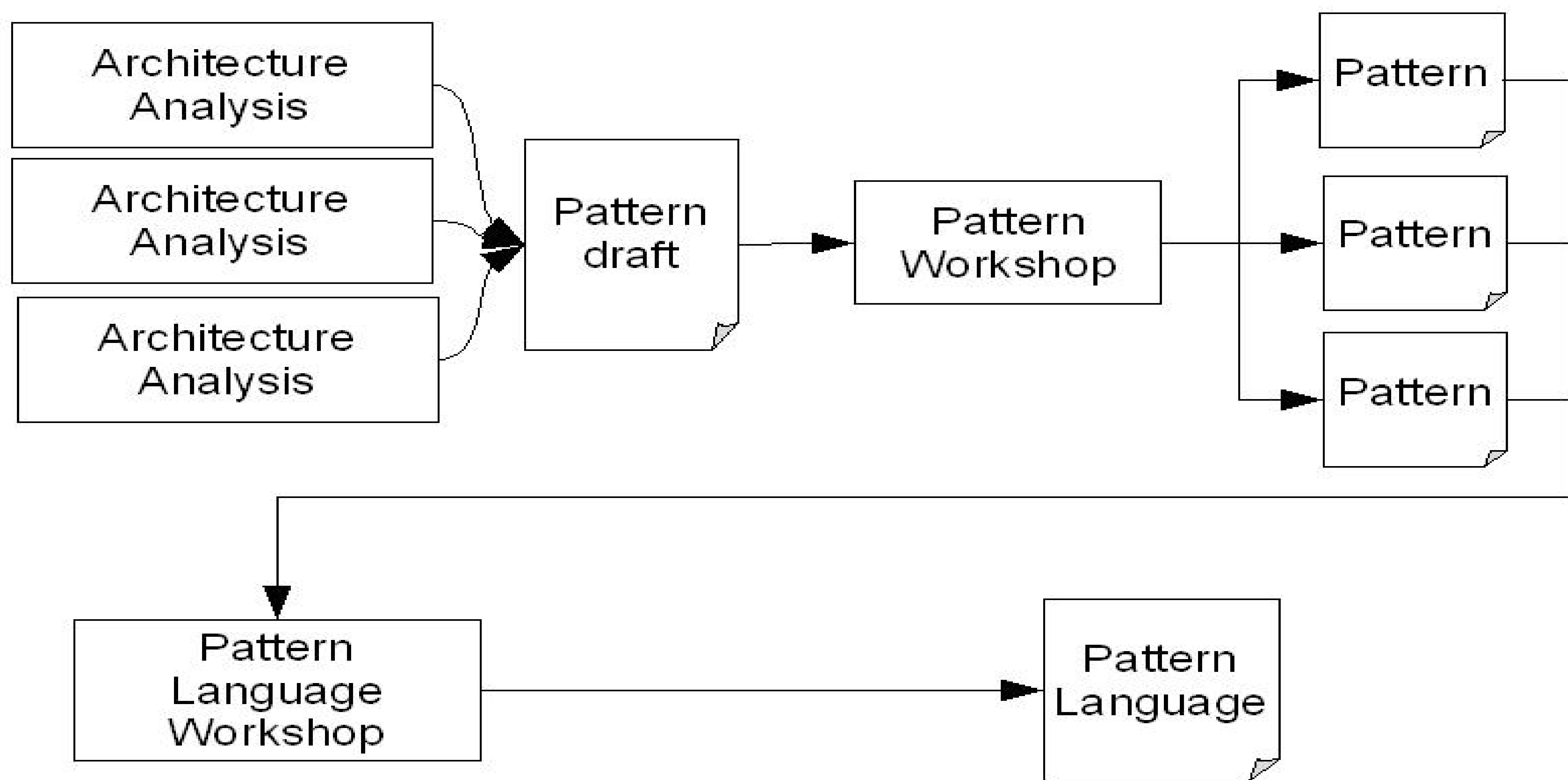
Background

- Sulake project: ATAM evaluations for machine industry during 2008-2009
 - Goal: evaluating embedded architectures and finding design patterns for them
 - Material: 4 companies, 2 analyzed systems per company
-

The problem

- Problem: collecting a comprehensive knowledge base for EMCS designers ► a method for collecting a pattern language
 - A pattern: a good solution to a recurring problem
 - Pattern language: a collection of patterns ordered so, that the designer can follow a logical path, "a sentence"
 - Old pattern collecting method: a sit-down
 - Collecting patterns from domain experts needs time and expertise
-

Our method



Phase 1: Identifying the patterns

- Add a new ATAM role: the pattern scribe
 - Find the best practices: seen at least in two systems
 - The pattern must improve the design
 - Not all common solutions are good ones
 - The evaluation team needs some domain expertise
 - Pattern mining is "free" work, as a analysis side-effect
 - Quality attributes help to connect the patterns
-

Example proto-pattern

- To ensure high availability and a short response time, the controlling unit is triplicated. In addition, the system contains a repeater, which selects the output of the controlling units based on heartbeat. All the units are operating, but only one output is really used to control the system.
-

Pattern workshops

- Naming the patterns
 - Using a pattern template (i.e. Giving the pattern a context, formulating the problem, identifying the forces...)
 - Several workshop iterations
 - Each pattern has an owner who is perfecting the pattern description.
 - Workshopped patterns are ready to be published
-

Language building

- Connecting the patterns together so that they generate the system design
 - Patterns generate or improve the entities in the domain
 - Resulting context of an earlier pattern forms the context to the next patterns
-

Results

- An example system
 - 25 fully analyzed scenarios
 - 11 patterns in total
 - One pattern as an architectural solution in 19 scenarios.
 - Other system
 - 5 patterns identified in the design by industry partner.
-

The end

- Questions?
-