DSVL: Function Block Language Lehman Laws

Mika Karaila,

Project Manager

metso

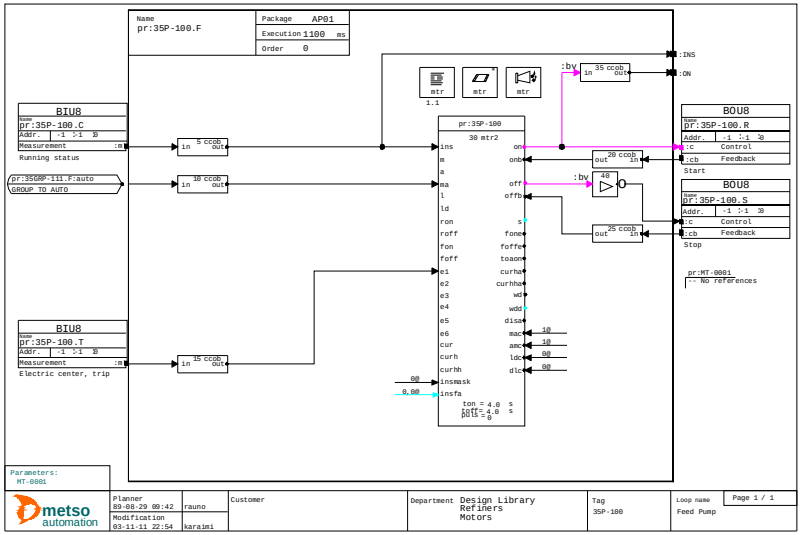
Outline

- Introduction to Function Block Language (FBL)
- Lehman’s laws and FBL
- Demo
- Summary & conclusions

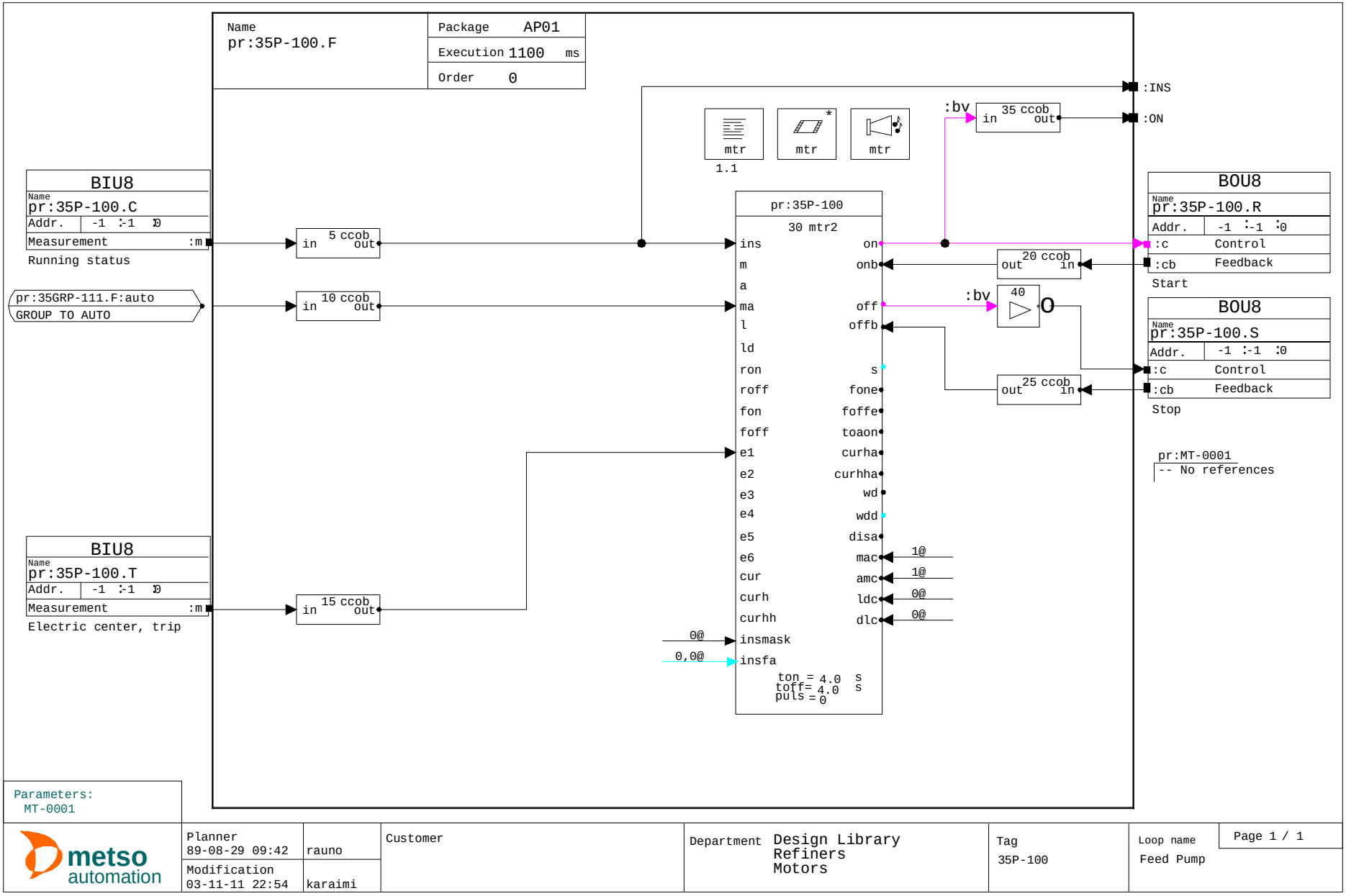
metso

Introduction to Function Block Language (FBL)

- Customer-specific process control solutions
- Function block program could be e.g. for controlling a water tank level
- Created with domain-specific visual language FBL:
  - Function Block CAD (FbCAD).
  - A function block is a capsulated subroutine. It runs functions according the given parameters and connections.
  - Each parameter value reflects to function block's functionality.
  - Connections bind dynamic values to a function block.
- Function block diagrams are compiled to executable byte-code for control system (real-time environment).
- An average customer project contains in a typical case 5000-6000 diagrams and over 20000 input/output-connections to the field devices.



FBL diagram example



Parameters:
MT-0001

metso automation

Planner 89-08-29 09:42	rauno	Customer	Department Design Library Refiners Motors	Tag 35P-100	Loop name Feed Pump	Page 1 / 1
Modification 03-11-11 22:54	karaimi					

Law I: Continuing change

- Operating systems: Unix, DOS, Windows
- For visual language: better graphics, faster to render, TrueType fonts (support for Unicode)
- Large displays: 640 x 480 -> 1024 x 768 or more

Solution: accept change and constant upgrades

metso

Law II: Increased complexity

- New function blocks (expected) => new symbols
- New IO cards (expected) => new symbols

- New field bus protocols (not foreseen) -> new “subdomains”, language level semantics and new “output” needed
  - Foundation Fieldbus Function blocks executed on device, not in DCS CPU unit => different kind of semantics needed
  - Profibus DP/PA, slave units capsulated as IO cards
  - Microsoft OPC, server/group/item capsulated as IO cards

Solution: Capsulation and abstraction

metso

Law III: Self regulation

- Operating systems are changing (Law I)
- New technologies coming (Law II)
- Customers are not able to take every year new release (factories running 24/7 and perhaps one planned shutdown each year)
- Critical systems require more testing and stability

Solution: Conservatism and longer upgrade interval

metso

Law IV: Conservation of organizational stability

- Architecture and teams, logical structures similar to Conway's law
- Takes time to be a talent programmer, requirements high

Solution: domain knowledge that requires multi-talented people will help in keeping organizational stability.

metso

Law V: Conservation of familiarity

- FBL symbolism and principles remained same from year 1989
- Logical operations same for all symbols

Solution: people are conservative and do not like very big changes it helps to keep things familiar.

metso

Law VI: Continuing growth

- Memory from 1-2 Mb to 1-2 Gb, bus speed 2 Mb/s -> 100 Mb/s
- Measurements from year 2000 -> 2008
  - FBL program size average function block amount 20->30
  - Project size from 1000 FBL programs -> 5000 FBL programs
  - FBL symbols 500 -> 1600
  - Code generator 36 kLOC -> 44 kLOC
  - DB adapter 21 kLOC -> 32 kLOC

Solution: flexible meta-model, architecture that separates extensions into symbols, language semantics and rules in compact code generator.

metso

Law VII: Declining quality

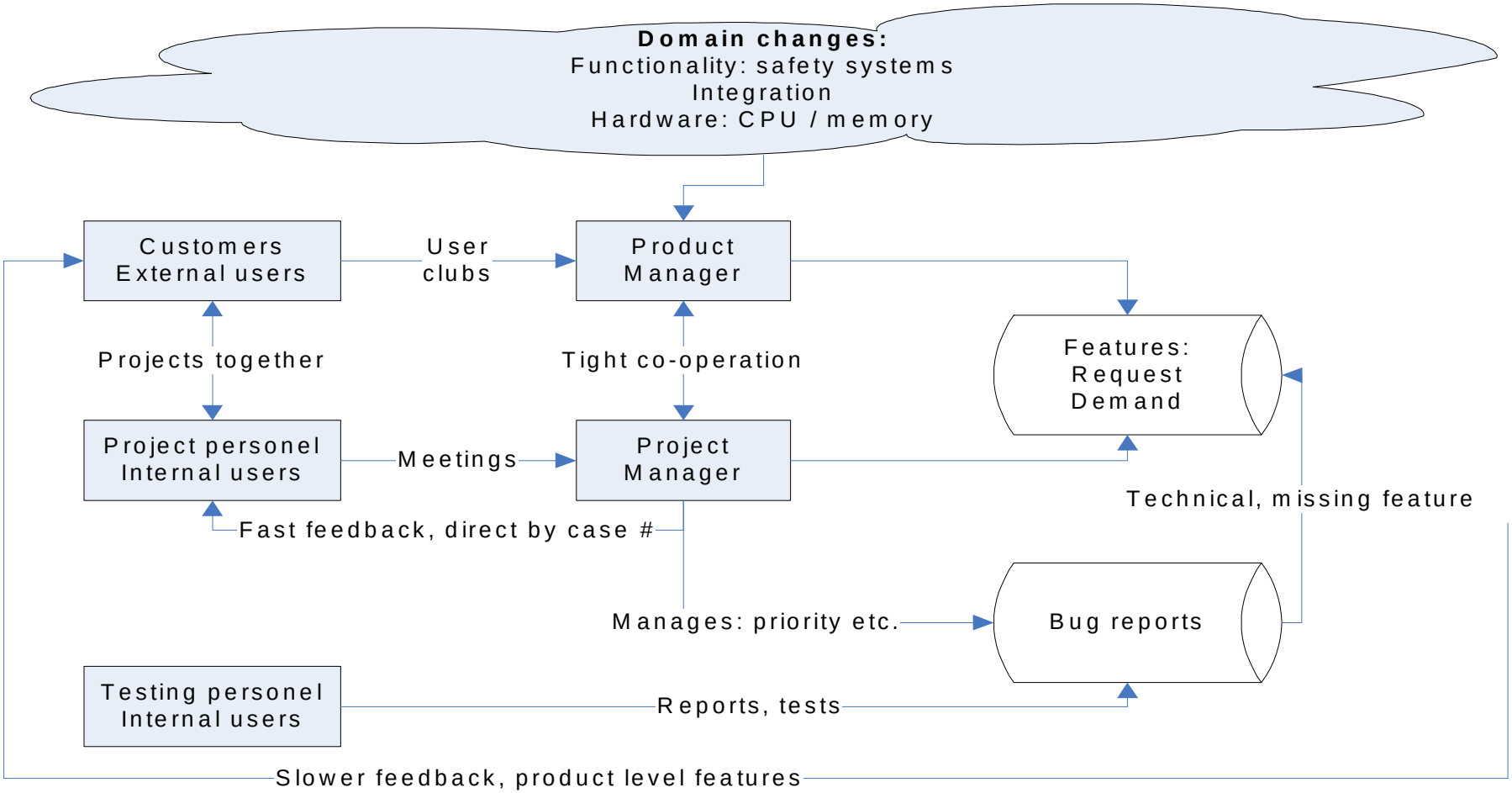
- Continuing change (Law I), increased complexity (Law II), continuous growth (Law VI) => quality problems can arise

- Added more checks and warnings to prevent earlier errors

Solution: FBL interoperability and compatibility used in regression testing to help in quality assurance.

metso

Law VIII: Feedback system



Law VIII: Feedback system

- FBL function block improvements
- FBL editor features: navigation, context sensitive menu
- FBL is dynamic, living language

Solution: formalized feedback system with feedback channels

metso

Patterns & idioms in Domain Specific Language

- FBL templates are forming patterns:
  - Basic functionality (core template)
  - Additional features in own template: interlocking, start/stop automation (feature template)

- FBL itself contains small idioms:
  - NOT implemented with XOR
  - Alarm masking
  - More existing, takes time to identify

metso

Demo

- FBL symbols
- FBL template
- FBL idiom

metso

Summary & conclusions

- In a dynamic environment, it is very important to manage the maintenance and evolution processes.
- Control the maintenance and evolution process with iteration. Feedback handling mechanism: priorities and new ideas for further development.
- Architecture is still dynamic and flexible.
- The management of development and maintenance processes help in evolution.

metso

Comments and questions ?

metso