Domain-Specific Language Architecture for Automation Systems: An Industrial Case Study

Christopher Preschern
Andrea Leitner
Christian Kreiner

Agenda

- My field of Research
- Motivation of the Paper
- What is PISCAS?
- PISCAS Demo
- PISCAS Architecture
- Results
- Conclusion

My field of research

- Embedded System Architectures
  - Safety
  - Security
- Architectural Patterns
  - Effect on Safety/Security Certification

→ this paper is based on results on results of my master‘s thesis

Motivation

- Suggest design decisions on how to easily develop DSLs for automation systems
- Show the application and the consequences of these design decisions

What is PISCAS?

- PISCiculture Automation System
  - www.piscas.eu
- Product Line approach to develop fish farm automation systems
  - MetaEdit+ for DSL development
  - B&R automation devices

What is PISCAS?

PISCAS Model

- Automation Software
  - Hardware mapping
  - IEC 61131 source code
  - Visualization
- Configuration Files
  - Web portal SQL files
  - Network device configuration
- Documentation
  - Graphical overview
  - Wiring plan
  - List of parts
  - Labels for wiring closet

PISCAS Demo

- Documentation
- Automation Software
- Web Portal

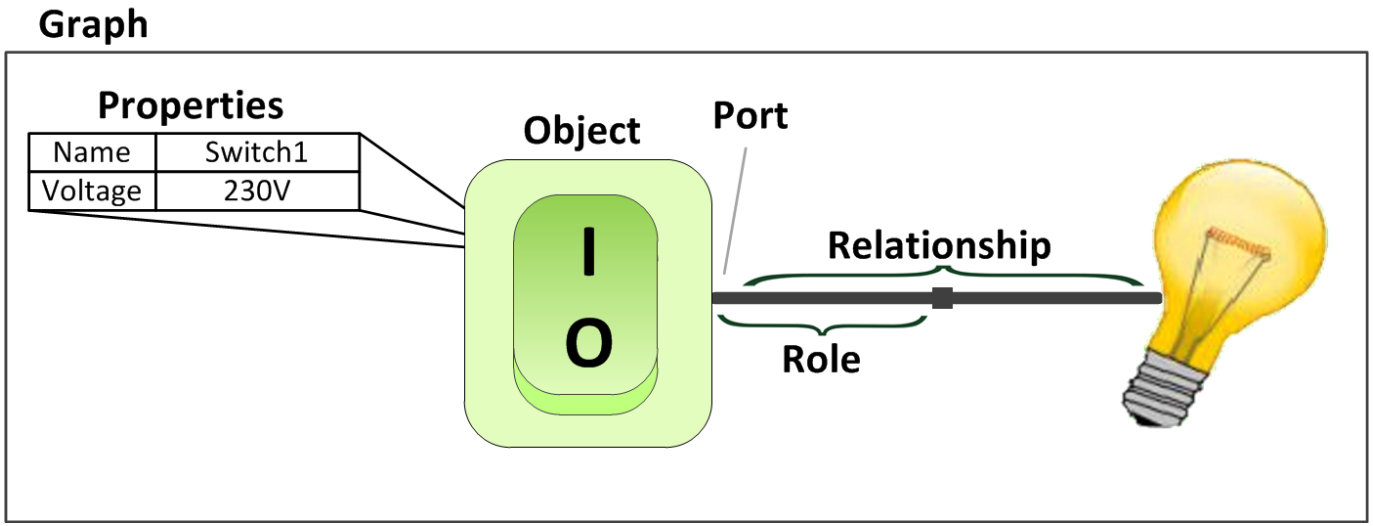
Fish farm automation domain description

- Elements: Feeders, oxygen level control, water level supervision + alarm system

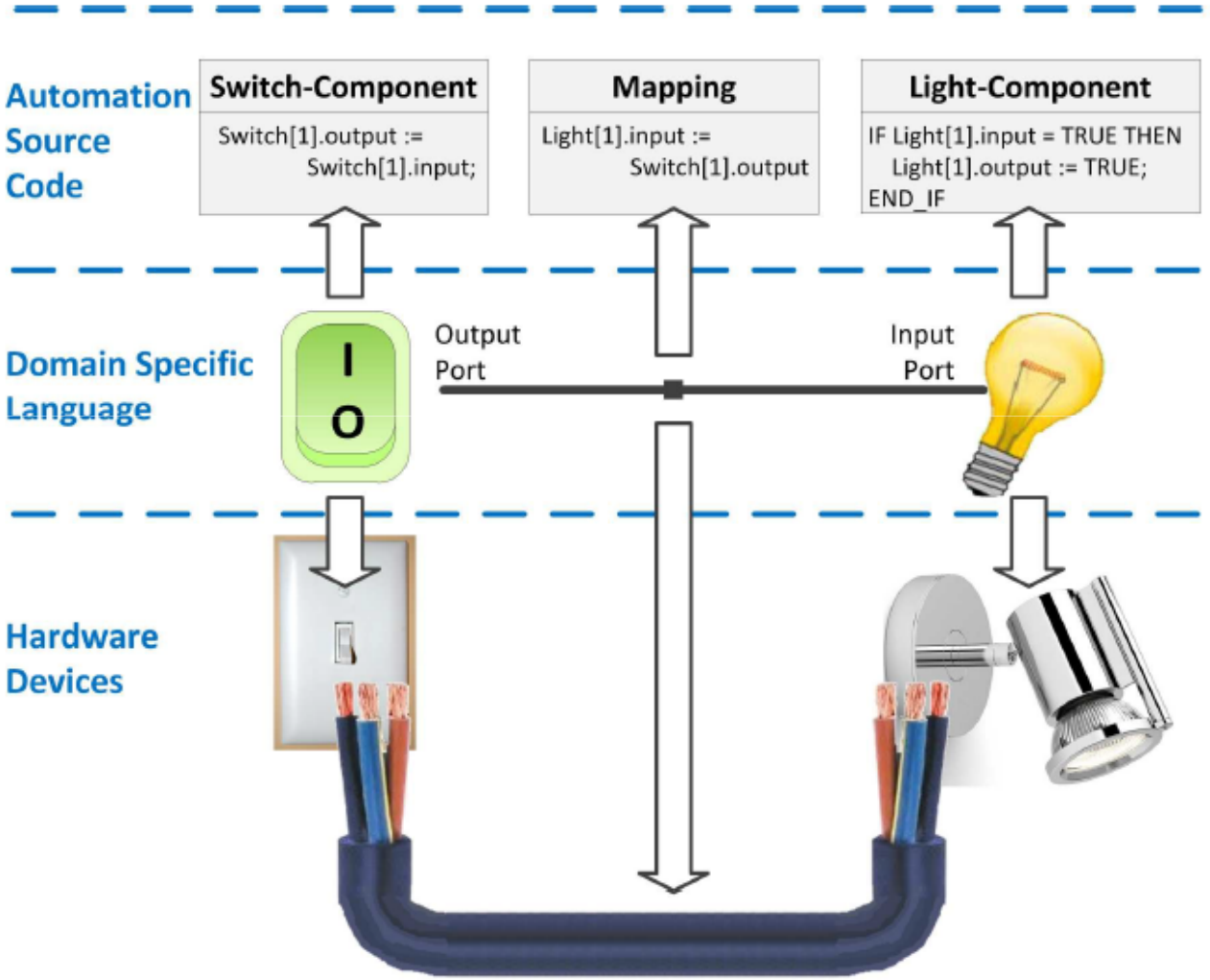
- Fish farms mostly differ in their arrangement/amount of devices
- Device types change rather often
- Main functionality stays the same

PISCAS Architecture

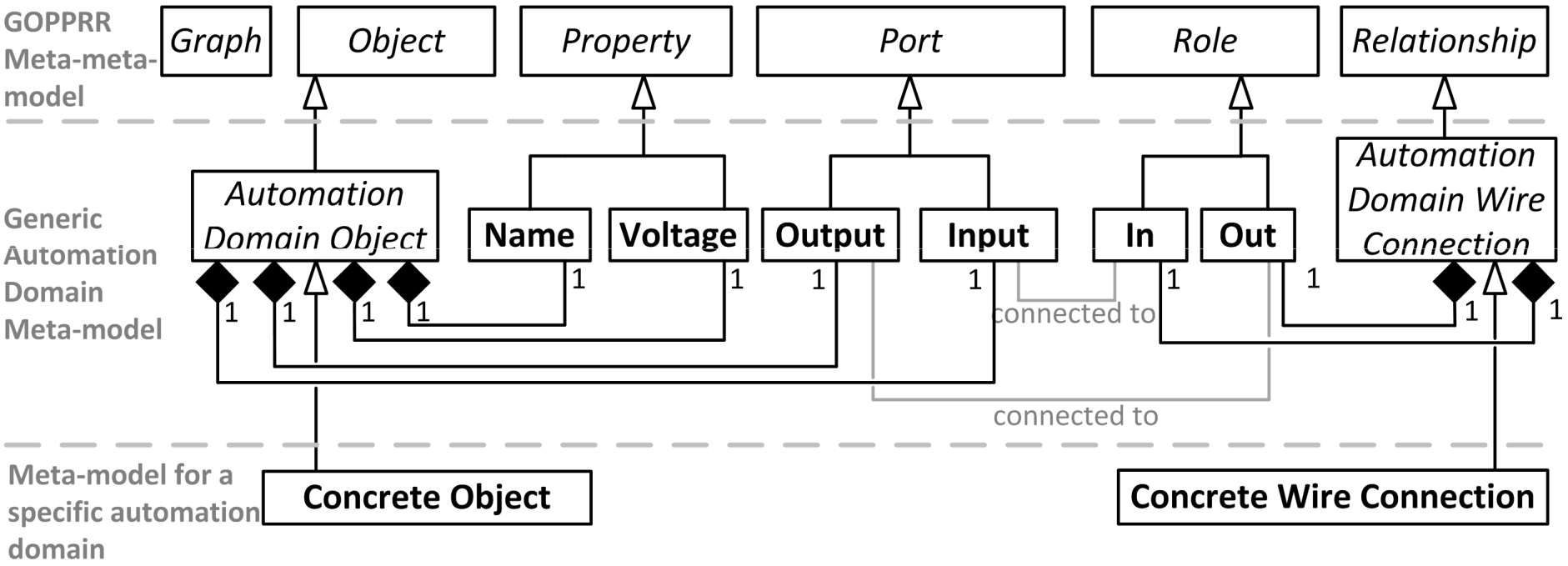
Physical system	GOPPRR concepts	Automation software
automation plant	*Graph*	overall software
device	*Object*	function block
wire	*Relationship*	connecting function block interface variables
-	*Role*	-
wire connection	*Port*	function block interface variables
device attribute	*Property*	function block parameters



PISCAS Architecture



PISCAS Architecture



PISCAS Design Decisions

- Abstract Object / Abstract Relationship
- Elements connected in Pipes&Filters style
- Direct representation of physical automation devices as Objects
- Explicit modeling of I/O modules

Benefits of the PISCAS Architecture

- Easy generation of documentation, visualization and automation software mapping
- Intuitive modeling for automation domain expert and fish farm expert

Final Remarks

- Provide guidelines for automation DSL development
  - benefits for visualization/documentation generation
  - benefits for application modeling
  - benefits for automation system maintenance
- Two PISCAS systems in operation
  - hardware installation plan was very useful
  - low software maintenance effort

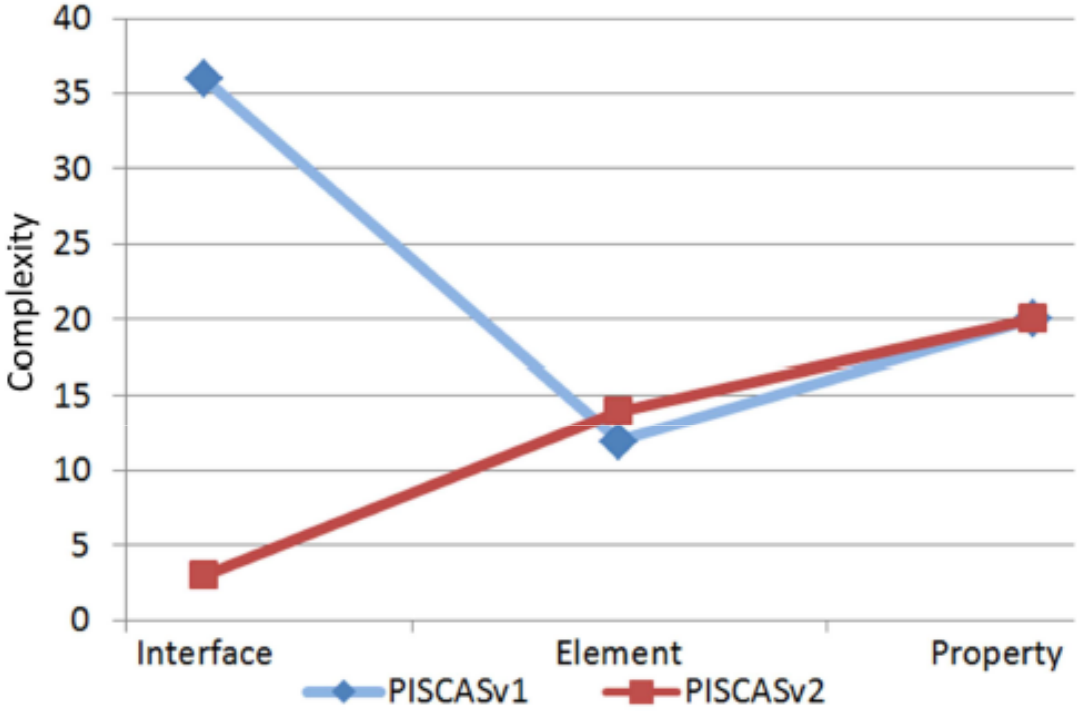
Thank you very much for your attention

Any questions?

PISCAS Application Modeling

	Fish Farm element modeling (ponds, switches, lights, ...)	B&R I/O module modeling
Fish farm A	2h	3h
Fish farm B	1h	1.5h
Add new components to B (model approximately doubled)	1h	2h

DSL Complexity

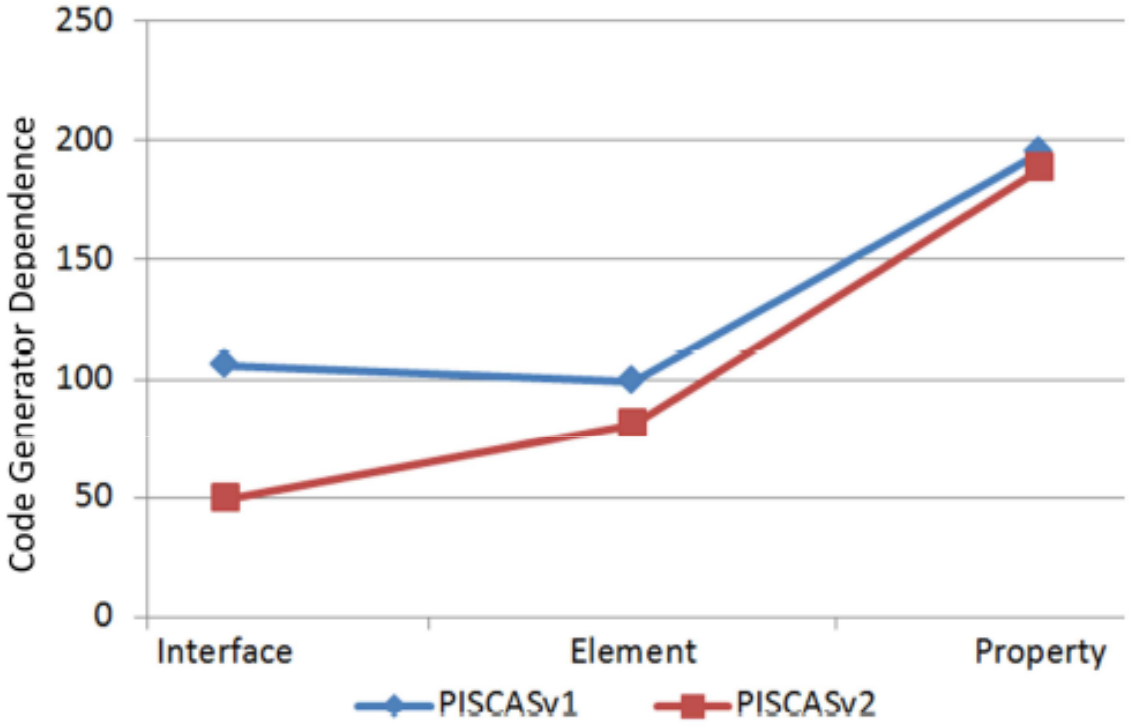


$$C_{interface} = n_{Relationships} + n_{Roles} + n_{Constraints}$$

$$C_{element} = n_{Objects} + n_{Ports}$$

$$C_{properties} = n_{Properties}$$

Code Generator Dependance



$$D_{interface} = \#LOC_{Relationships} + \#LOC_{Roles} + \#LOC_{Constraints}$$

$$D_{element} = \#LOC_{Objects} + \#LOC_{Ports}$$

$$D_{properties} = \#LOC_{Properties}$$