Engineering Environment Integration Across Disciplines with the Engineering Service Bus

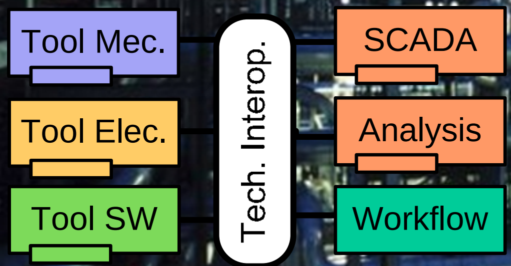
<u>Dietmar Winkler</u> **Stefan Biffl**

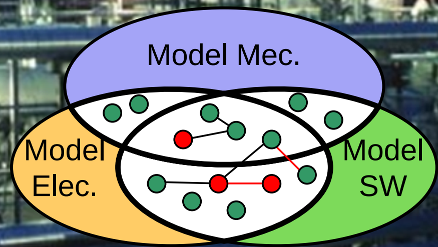
Christian Doppler Laboratory SE-Flex-AS

Institute of Software Technology and Interactive Systems (ISIS)

Vienna University of Technology

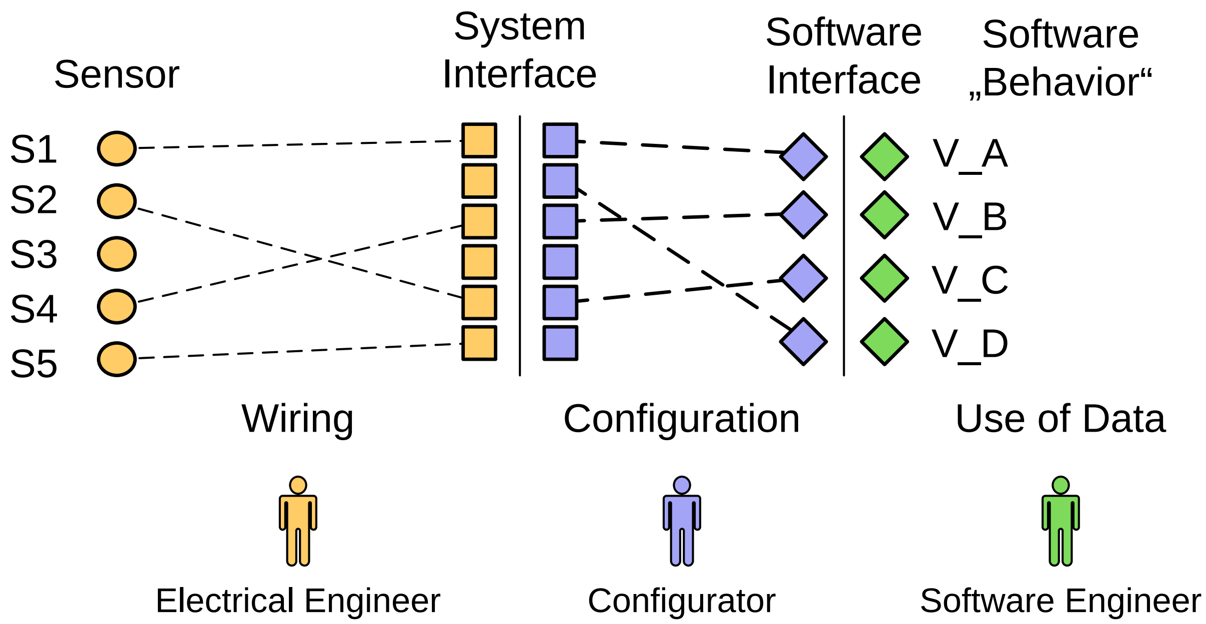
http://cdl.ifs.tuwien.ac.at

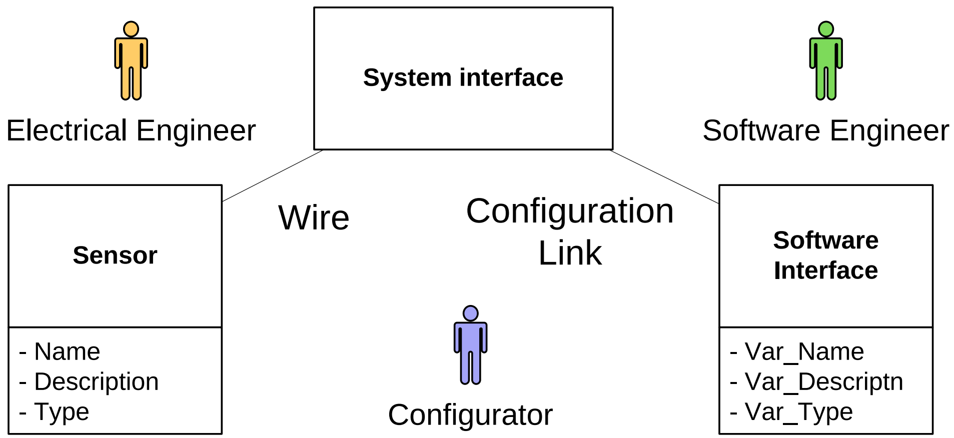




End-to-End Test Across Engineering Models

Use of common concepts in models across engineering disciplines



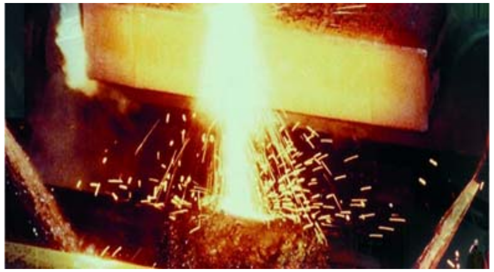


End-to-End Analysis

- List of sensor name/description/type with Variable name/description/type
- Warnings for incomplete chains between variables and sensors

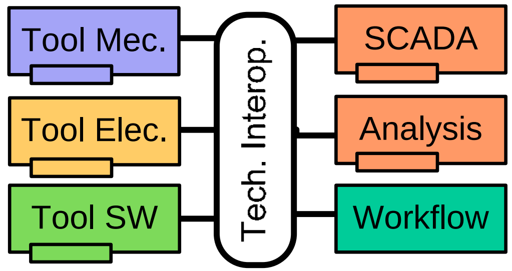
Motivation and Overview

- Software-intensive systems.
- Several disciplines cooperate in an industry environment.
- Engineering models (e.g., mechanics and electronics) contain requirements and design constraints for software engineers.
- Existing models and tools focus on supporting engineers in specific disciplines.
- Human experts bridge technical and semantic gaps between models and tools of different engineering disciplines.

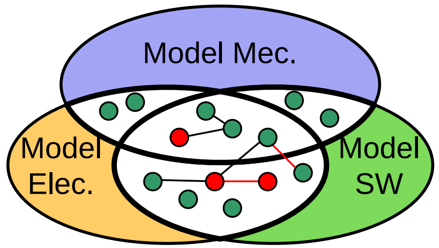


Scope of Research

- **What hinders effective collaboration across disciplines?**
  - Domain- and vendor-specific solutions (e.g., point-to-point integration).
  - Heterogeneous models in various disciplines.
  - Different stakeholders and different "languages".
  - Limited connection between development and operation.



- **Concept evaluation based on real-world use cases and prototypes**
  - Technical Integration of Tools.
  - Semantic Integration of Data.
  - Quality Assurance across Engineering Disciplines.



- Christian Doppler Laboratory started on January 2010.

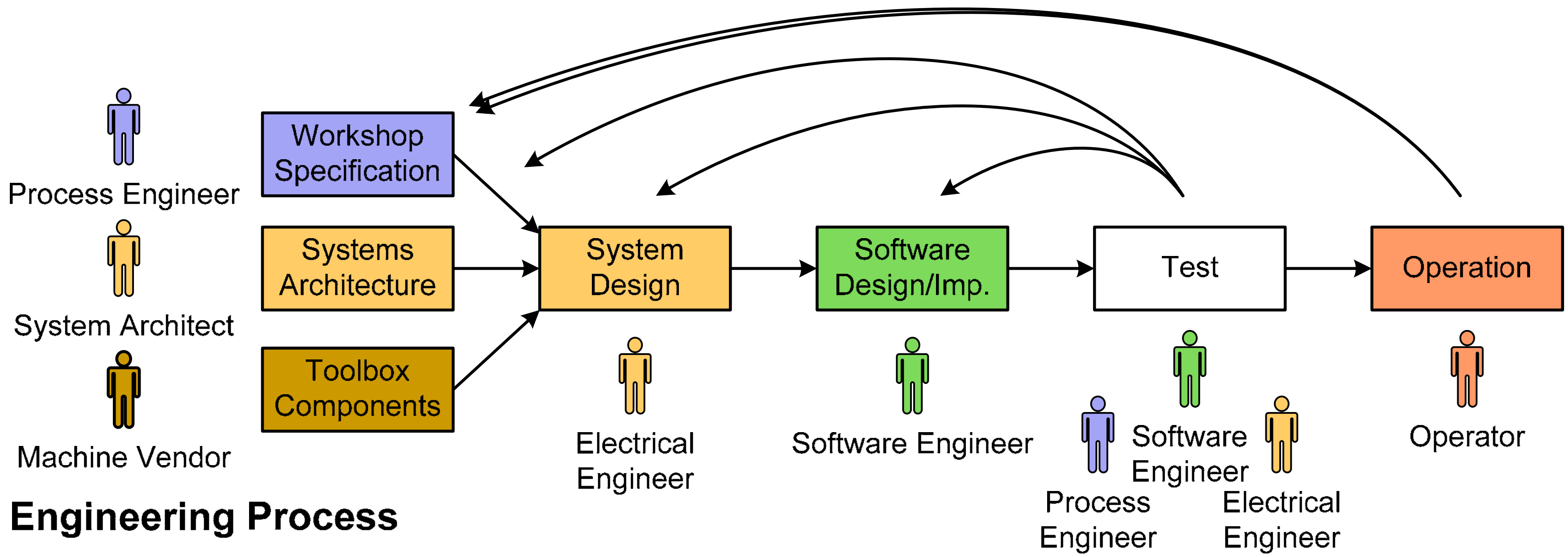
Software Engineering Integration for Flexible Automation Systems

Basic research challenges

- Early defect detection across engineering discipline and tool boundaries.
- Engineering process analysis using design- and run-time data sources.

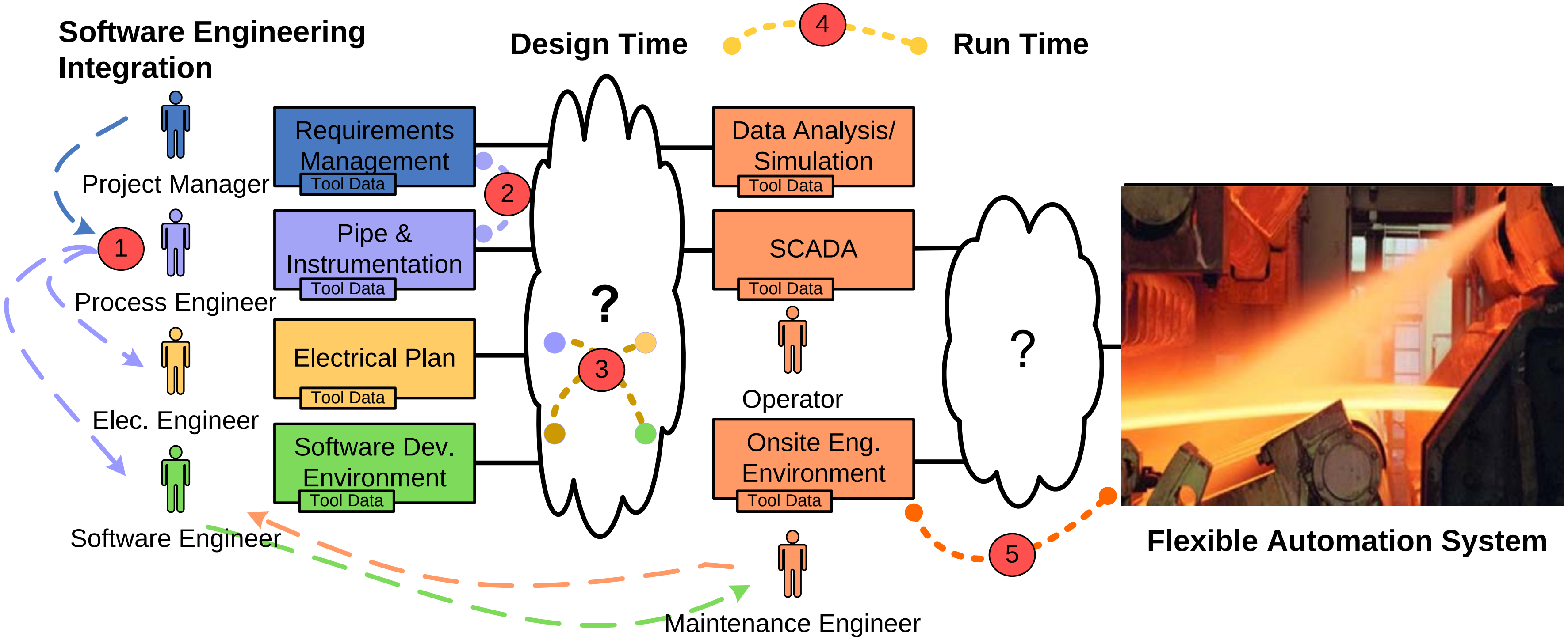
Research applications in the industry partners' domains

- Platform to build integrated tools for automation systems development & QA.
- SCADA systems with data analysis for monitoring automation systems.



Engineering Process

Challenges and Requirements

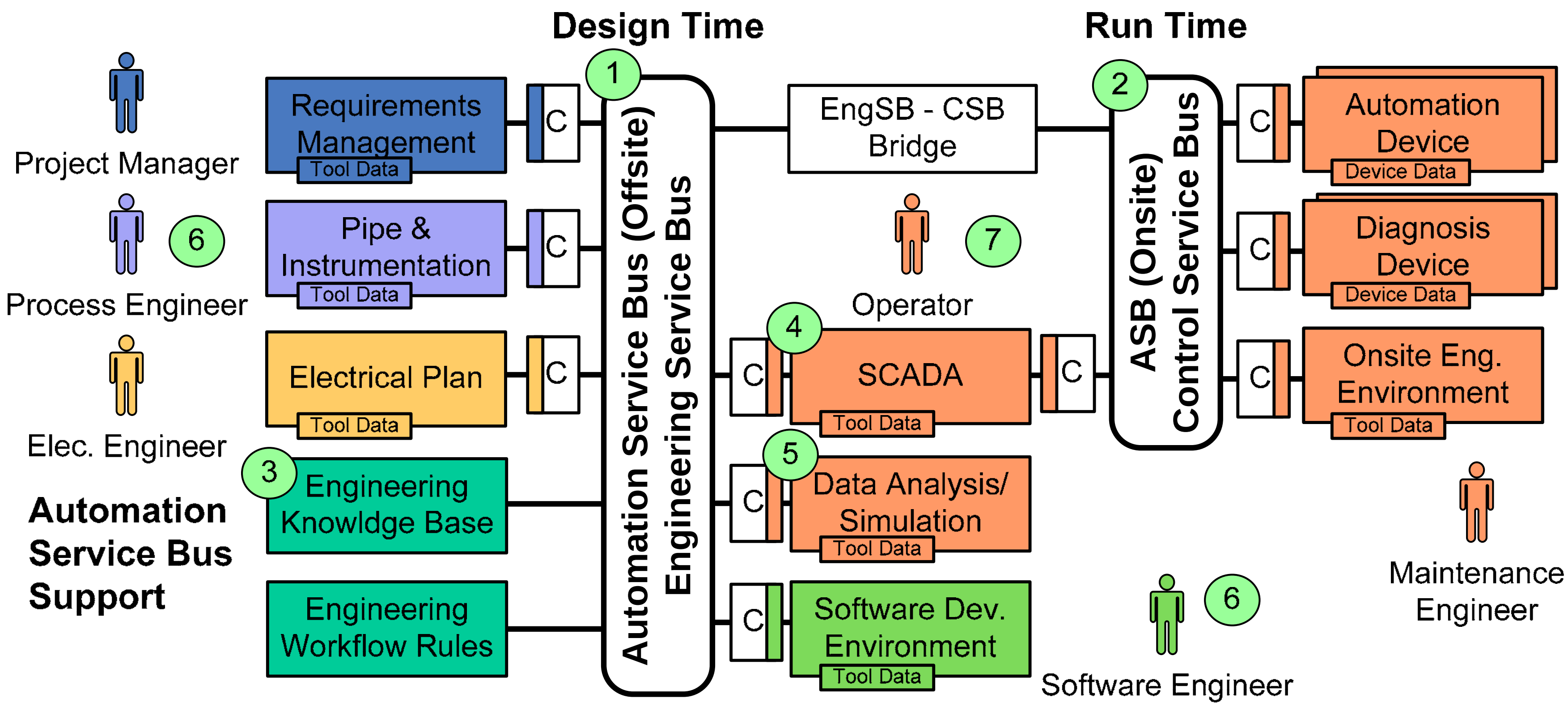


Challenges from weak integration of software tools for engineering

1. **Engineering process** on event level is hard to track and analyze.
2. **Integration of software tools** is often vendor-specific and/or fragile.
3. **Sharing of data models** across software tools is inefficient and risky.
4. **Run-time defect detection** cannot easily access design knowledge.
5. **Integration of run-time environments** is hard to observe for analysis.

Automation Service Bus

Goal: Approaches for the integration of software tools in automation engineering.

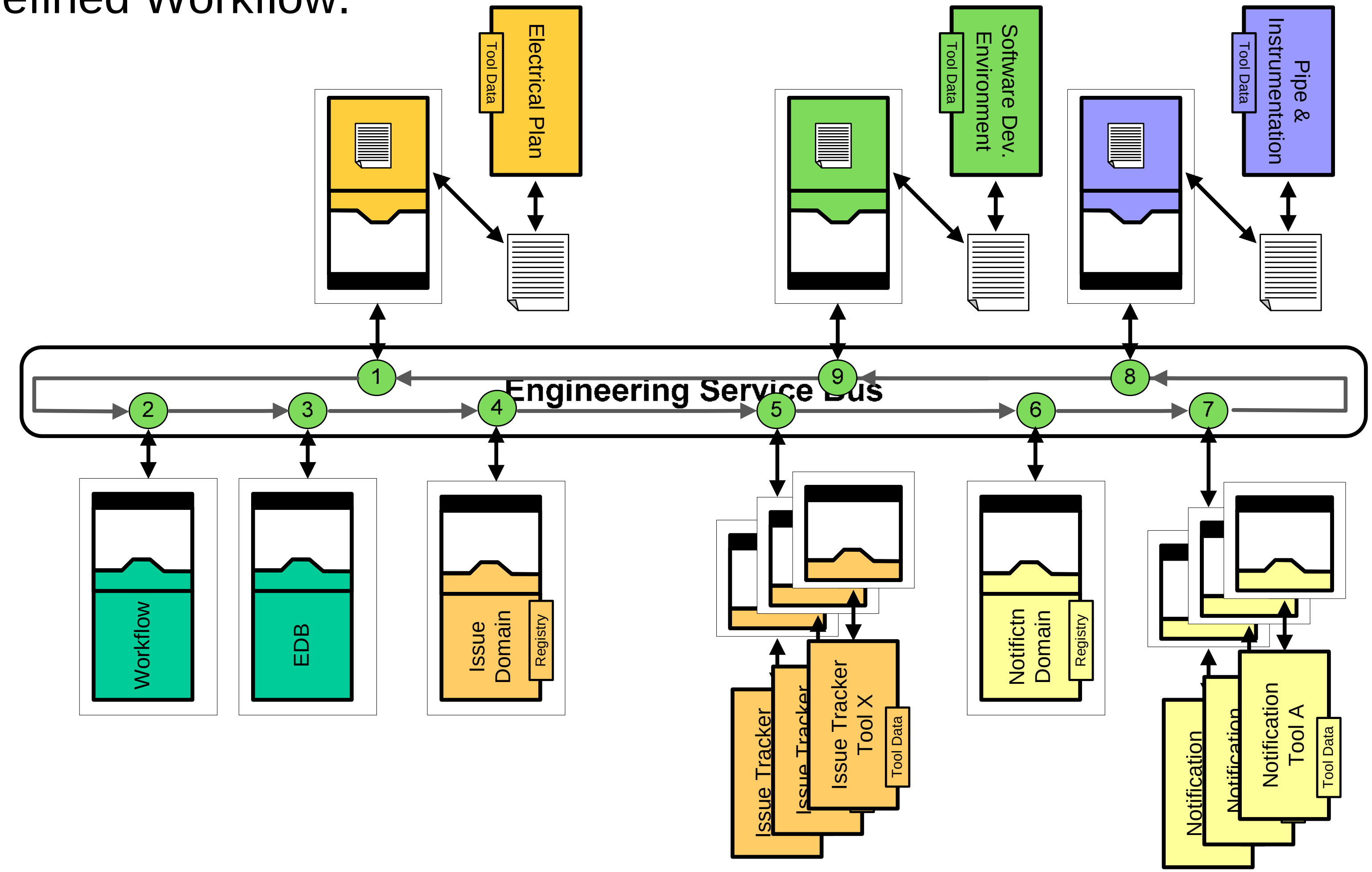


- Technical Integration: Engineering Service Bus (1), Control Service Bus (2).
- Semantic Integration: Engineering Knowledge Base (3).
- Flexible integration of SCADA (4) with data analysis/simulation (5).
- Defect detection approaches for design time (6) and run time (7).

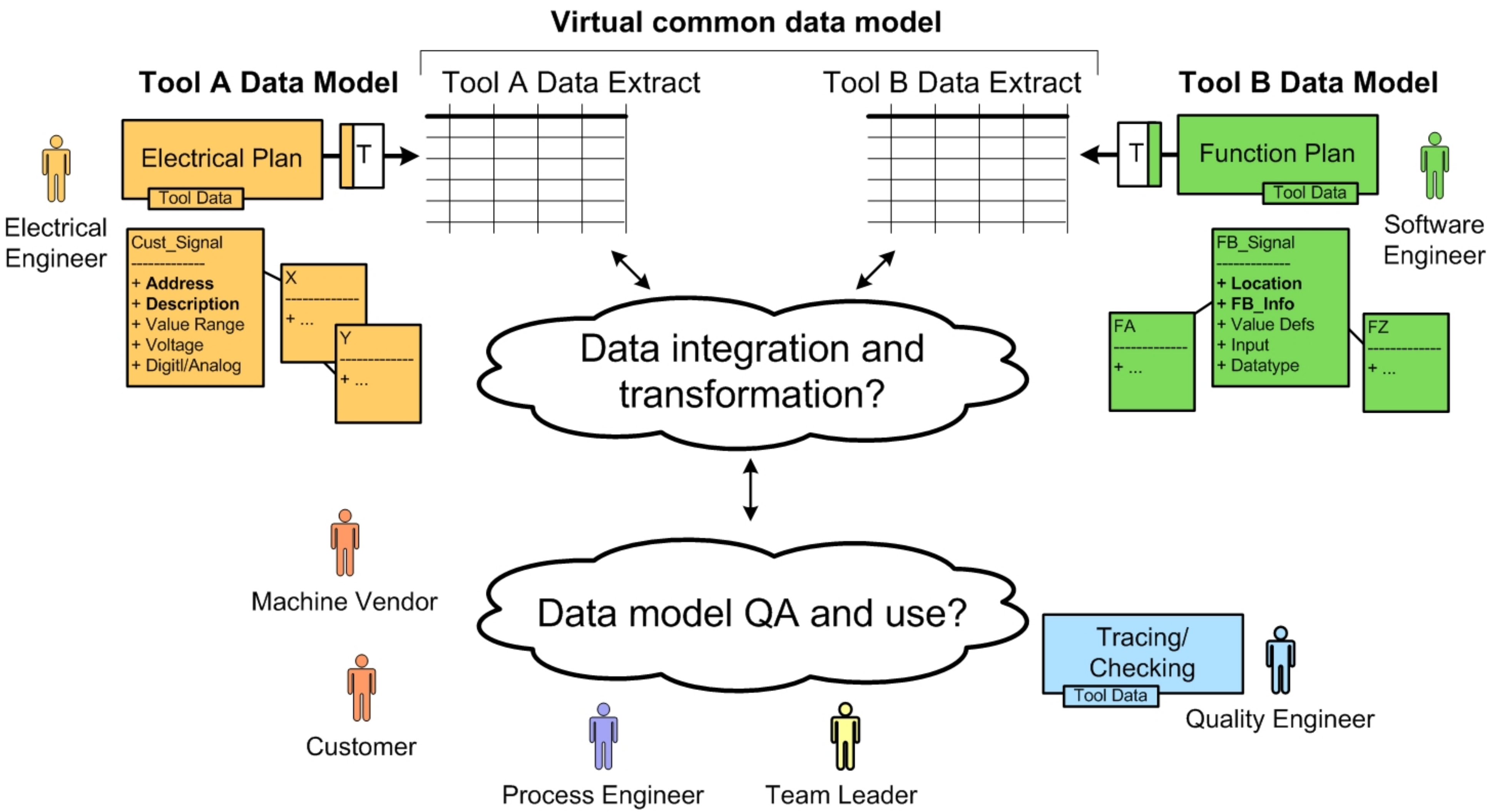
Example: Technical Systems Integration and Interoperability

Approach for integrating available automation engineering tools

- 3 heterogeneous engineering tools.
- Defined Workflow.

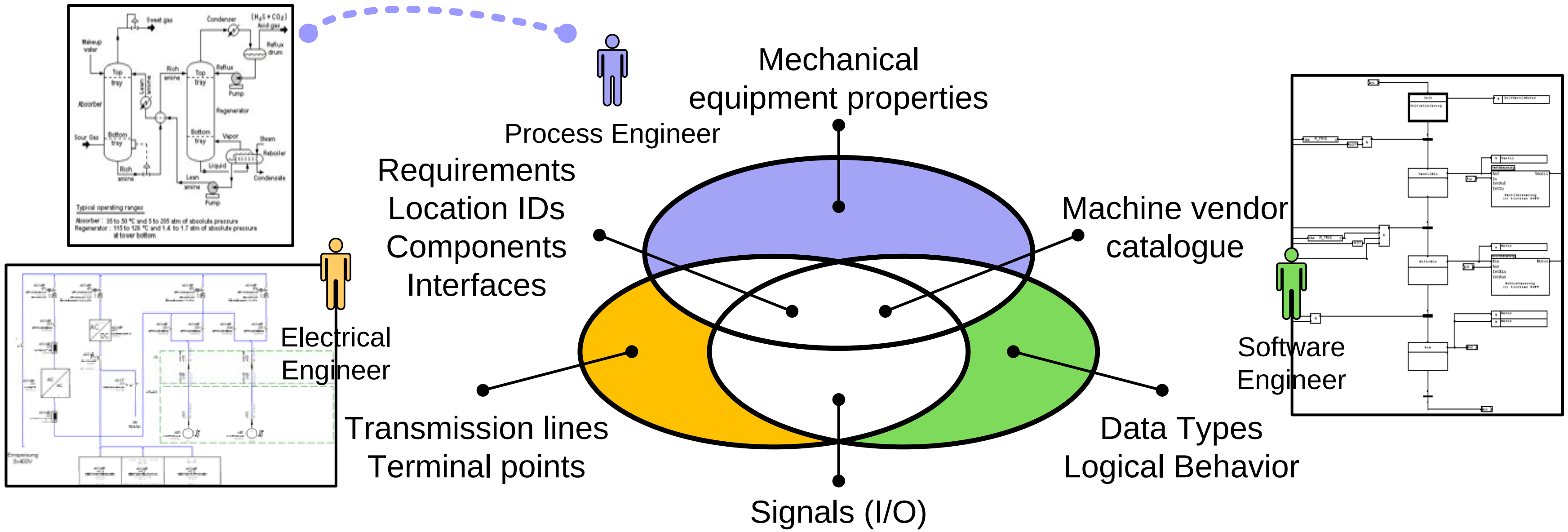


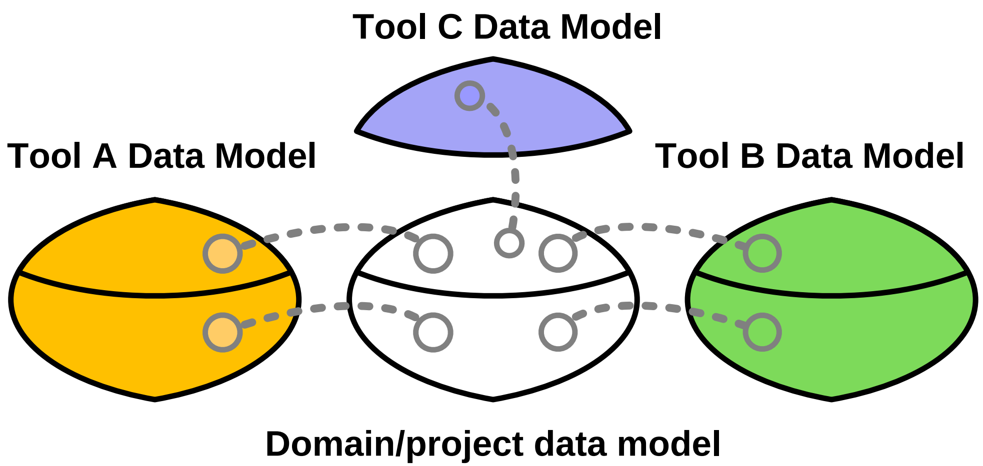
Collaboration Across Disciplines: Semantics?

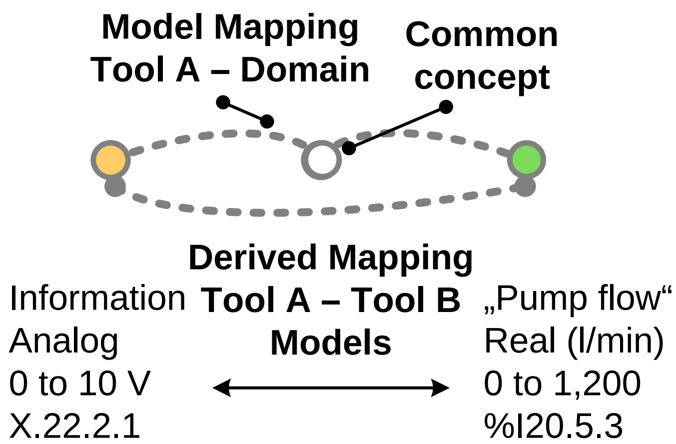


Semantic Integration of Engineering Knowledge

Identification of common concepts across engineering disciplines.

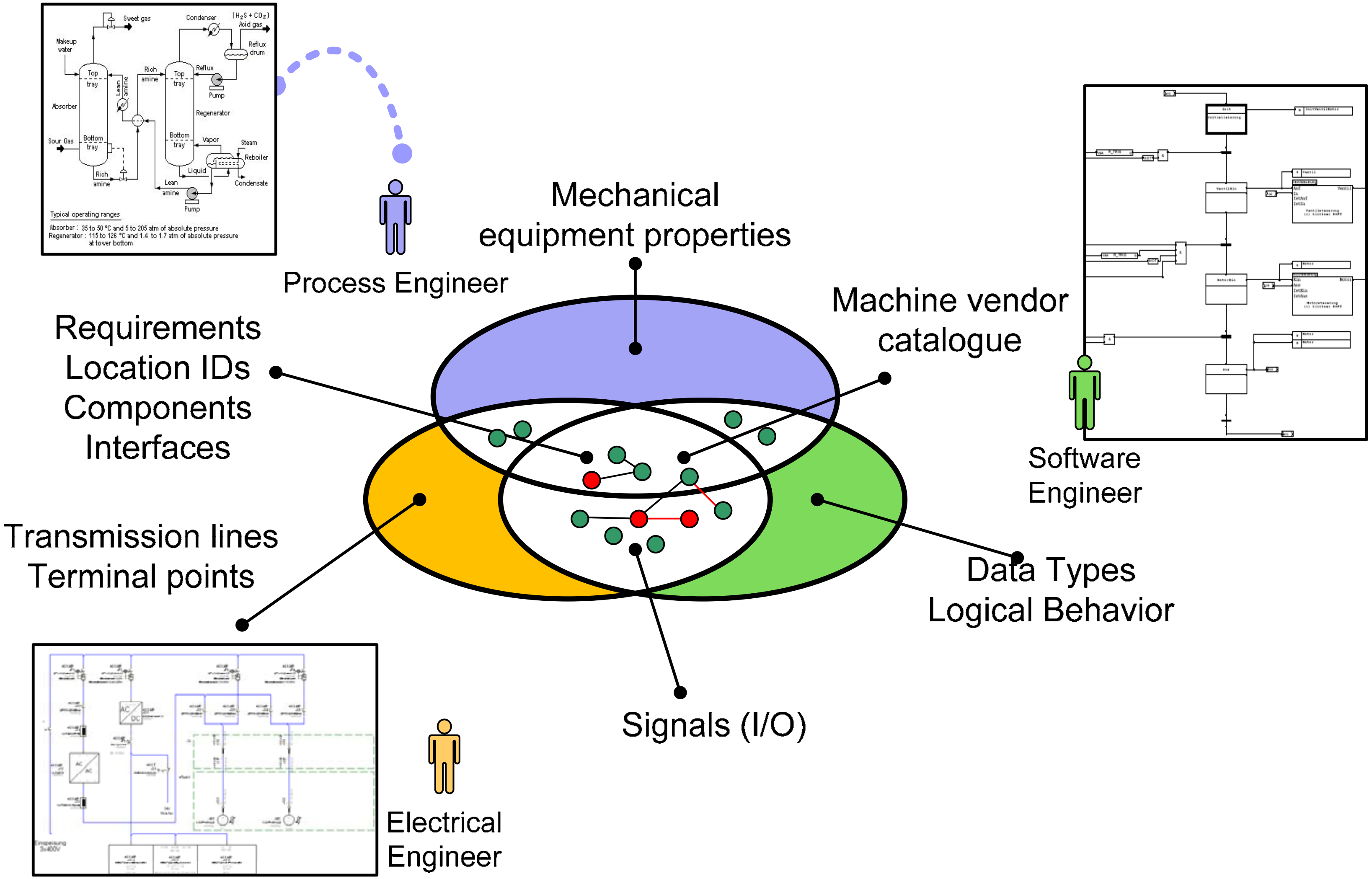






Defect Detection Across Tool Boundaries and Disciplines

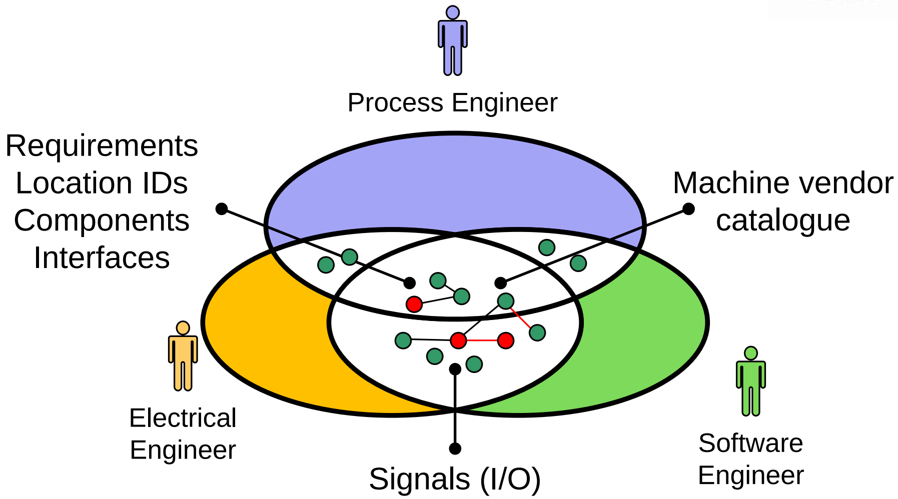
Use of common concepts in models across engineering disciplines



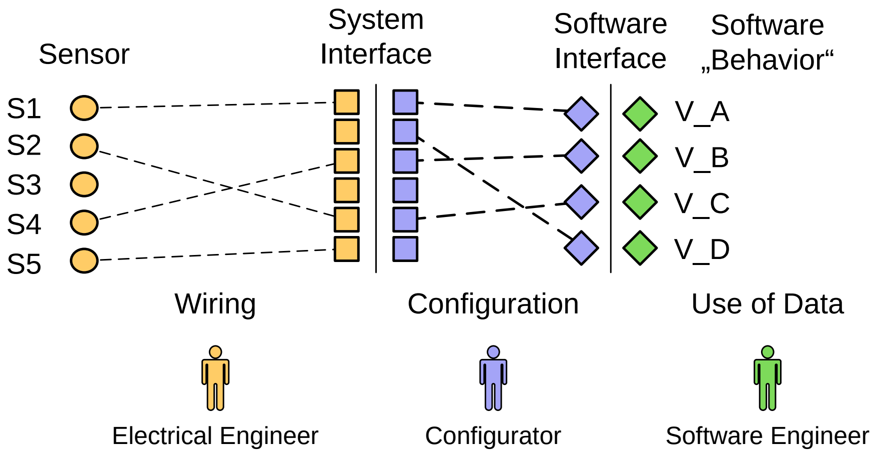
End-to-End Quality Assurance

- Challenge: Defect Detection across engineering disciplines

- **Identification of various defect types:**
  - Missing, wrong, inconsistent model elements or relationships.
  - Conflicts from changes to overlapping model elements.
  - Run-time violation of model constraints.

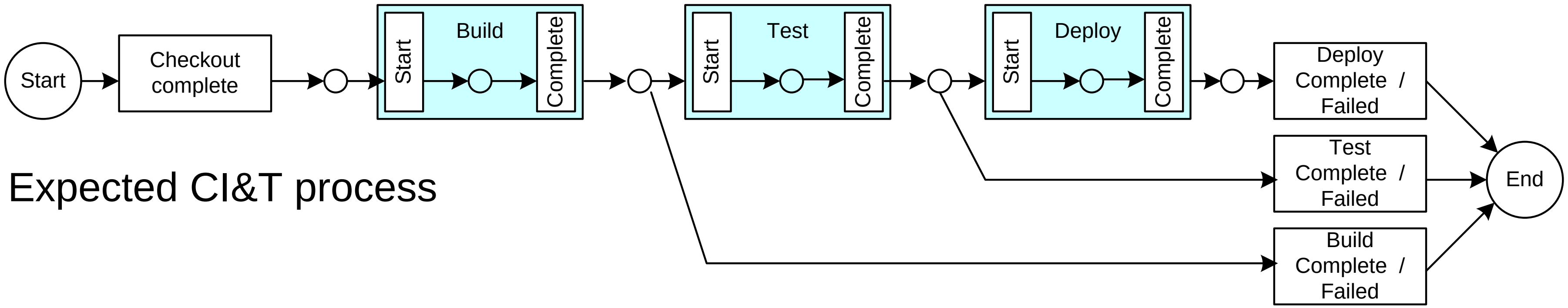


- **Quality Assurance approaches**
  - Review of overlapping model parts, e.g., with inspections.
  - Automated check of model assertions (syntactic and semantic).
  - Change conflict detection and resolution.

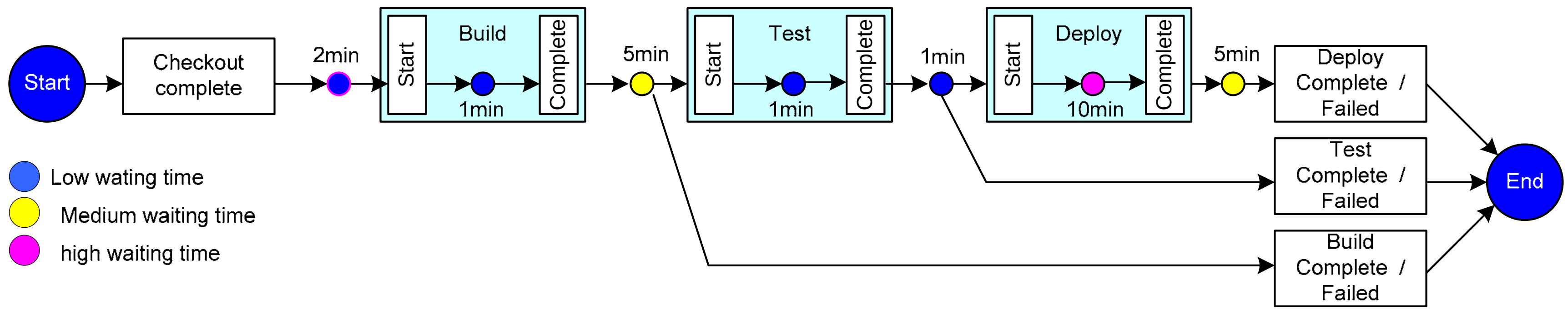


Engineering Process Analysis (CI&T)

- **Process automation, analysis and assessment** based on (EngSB) event logs
  - Visualization of the expected engineering process.
  - Comparison of expected with traces of actual engineering processes.
  - Analysis of actual engineering process variants (frequency of paths taken).
  - Measurement of engineering process duration, waiting and execution times.
- Example: Continuous Integration and Test (CI&T).



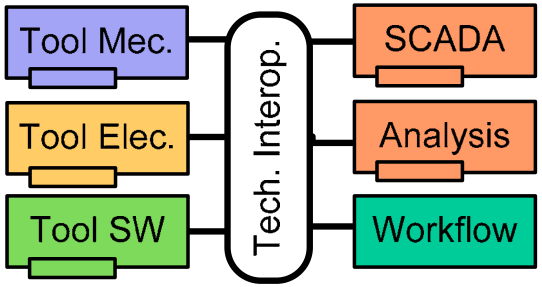
Process analysis based on sample engineering logs..



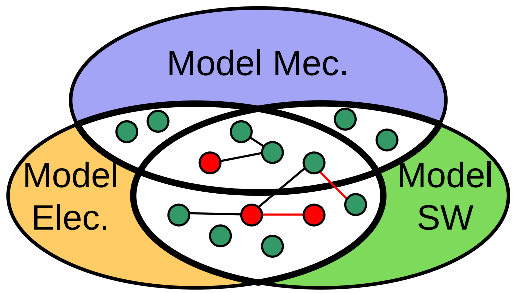
Summary

- Multi-disciplinary engineering projects are prone to risks from defects and delays due to technical gaps between tools and semantic gaps between data models.
- Technical and semantic integration provide the foundation for engineering process automation and quality management to lower these project risks.

- **The Engineering Service Bus (EngSB) environment provides:**
  - Technical Integration: Workflow-Rules and Events.
  - Semantic Integration: Data Models across disciplines.
  - Defect Detection & Engineering Process Automation: Engineering rules and process analysis.



- **Industry Use Cases**
  - End-to-End Quality Assurance.
  - Difference analysis between signal versions.
  - Defect detection in data models across tools and engineering disciplines.
  - Engineering Process Automation, Analysis and Improvement.



Thank you ...

Engineering Environment Integration Across Disciplines with the Engineering Service Bus

Dietmar Winkler
Stefan Biffl

Christian Doppler Laboratory "Software Engineering Integration for Flexible Automation Systems"
http://cdl.ifs.tuwien.ac.at

Institute of Software Technology and Interactive Systems (ISIS)
Vienna University of Technology

{Dietmar.Winkler, Stefan.Biffl}@tuwien.ac.at