Christian Doppler Laboratory

Software Engineering Integration
For Flexible Automation Systems

Support for the Multi-Disciplinary Engineering of CPPS
with Collective Intelligence Systems

Stefan Biffl  Angelika Müsir  Marta Sabou
Institute of Software Technology and Interactive Systems
Vienna University of Technology
http://cdl.ifs.tuwien.ac.at
Cyber-Physical Production System (CPPS)
Example Flexible Production System at University Magdeburg

- **Production System** elements
  - Machines with several tools
  - Transport and storage system with conveyors and turn tables

- **Engineering Process V-Model (VDI 2206)**
  - System Design
  - Domain-specific engineering
  - System integration
  - Validation of system properties
CPPS Engineering Challenges and Research Needs

Challenges

- Added complexity from **variability at run time** (self-*, plug and produce)
- Change from stable hierarchical control to **flexible decentralized control**
- **Multi-disciplinary engineering** methods and terminologies
- **Heterogeneous data models** and tools

Research Needs to support the engineering process

- (a) **Vertical integration** along all levels of the automation pyramid
- (b) **Horizontal integration** along the product lifecycle (yellow arrows)
- (c) **Smart assistance** functions
- (d) **Coherence of virtuality & reality**
Support for Knowledge-Intensive Tasks/Capabilities in CPPS Engineering

Knowledge-Intensive Task Examples
- Searching, e.g., for expertise, reusable artifacts
- Classification of artifacts
- Checking artifacts for defects
- Recommendation of artifacts for reuse

Knowledge-Intensive Capability Examples
- Formation of community of practice
- Awareness of expertise to enable expert search
- Knowledge management for reuse of artifacts
Why are KITs Challenging in CPPS Engineering?

- How to **coordinate the communication** of distributed, engineering teams?
- How to **aggregate distributed, local engineering knowledge** between multi-disciplinary engineering teams?

Complex knowledge fragmented across teams

Information sharing hampered by use of heterogeneous communication channels

Multiple & Distributed engineer teams

Implicit knowledge
Collective Intelligence (CI)

Group intelligence that emerges from collaboration, collective action, and competition of many individuals.
Collective Intelligence Systems (CIS)

Socio-technical systems that enable collective intelligence.

• effective, bottom-up information creation, modification and sharing
• continuous flow of user-generated content
• growing repository of valuable knowledge/data.

How can CIS support CPPS Engineering?

- **Effective coordination and sharing** of relevant know-how
- **Improved aggregation and discoverability** of distributed knowledge
CIS Example: GitHub

- Collaborative, global source code repository platform.
- Strong, diverse community.
- Example of an active, sustainable system.

Mark and build upon

Analyze forks and contributions

Explore new projects

Discuss topics and issues

Analyze activity level

Analyze project progress

Connect with developers
Success and Risk Factors

Success Factors:
- Designing the “right” system.
- Provide **low friction, easy to use means on contributing content**.
- Effective feedback mechanisms, which make users aware about activities of other users.
- Fostering an active community of contributors through engagement mechanisms (incentives).

Risk Factors:
- If CIS is **not well integrated** in daily users’ workflow routine, the CIS **will not be used**.
- Unbalanced focus on **social aspects** and **technical aspects**
- Handling of security and **privacy of user data** (logs).
Promising Field for Future Research

- Investigation of CIS application in Industry 4.0
  - How to support and improve multi-disciplinary engineering processes with CIS?
  - What are CPPS engineering-specific aspects relevant for CIS architecture design?

- What are typical software architectures used in successful CIS?
  - Can we formalize architectural knowledge in more detail using established tools and methodologies?
  - Can architectural patterns observed in CIS at large be applied in CPPS design settings?
Summary

- **Knowledge Intensive Tasks frequent in the design of CPPS**
  - Require access to global engineering knowledge
  - Engineering knowledge is fragmented, distributed and implicit

- **Collective Intelligence Systems:**
  - Provide bottom-up, self-organization capabilities to large, distributed groups with respect to information aggregation, organization and distribution.

- **CIS in CPPS engineering:**
  - Coordination of communication, information sharing and knowledge aggregation