

**AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE**

**AGH UNIVERSITY OF SCIENCE  
AND TECHNOLOGY**

**AGH**

# Design and verification of Smart City ultra complex systems

September 4-6, 2019



## CPS driven Smart City\*

Cyber Physical Systems (CPS) represents integration of physical and embedded systems with communication and IT systems.

Systems of Systems (SoSs) are related to evolving large-scale systems and coordination among those systems focusing on integration and optimization to satisfy given criteria. The concept is independent of the type of system.

**SoSs may incorporate CPS, but may be themselves considered as CPS\*\*.**

\*Houbing Song, Ravi Srinsivasan, Tamin Sookkoo, and Sabina Jeschke: Smart Cities. Foundations, Principles, and Applications, Wiley, 2017

\*\* Houbing Song, Danda B. Rawat, Sabina Jeschke, and Christian Brecher: Cyber-Physical Systems. Foundations, Principles, and Applications, Elsevier, Intelligent Data Centric Systems, 2017



# Cyber-Physical Systems

Selected features:

1. Distributed consensus control for wireless CPSs
2. Online control and optimization for CPSs
3. Energy-harvesting low-power devices
4. Machine2Machine communication over 5G
5. Data reliability
6. Network-wide programming
7. CPS – human interaction
8. Security and privacy
9. ...

## Development of CPS

CPS components are related to different domains and are modelled using variety of formalisms/languages.

### Questions

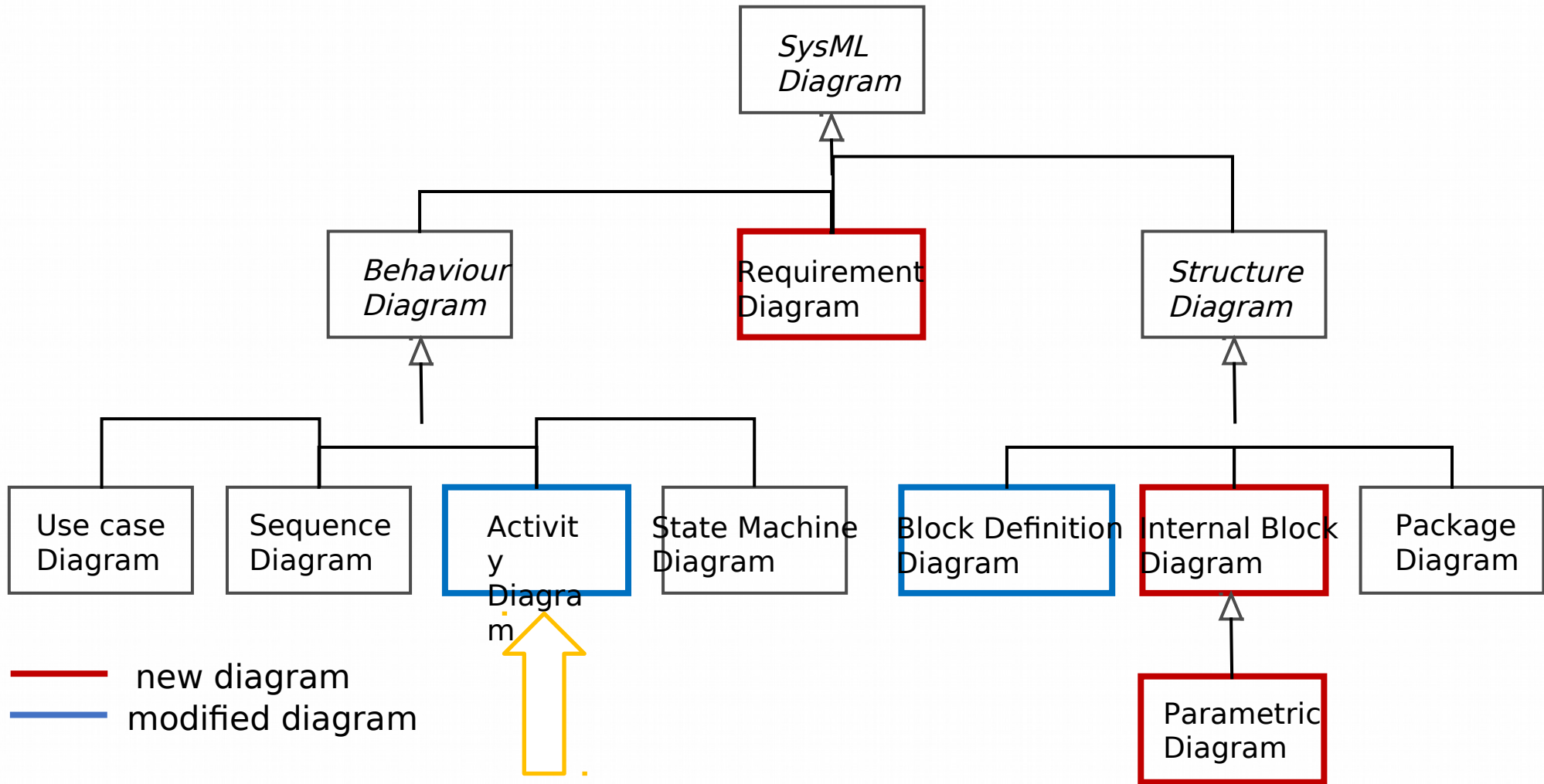
- 1. Is it impossible to find common language for modeling?**
- 2. May we use CPS development life-cycle as a framework for development of smart city systems?**



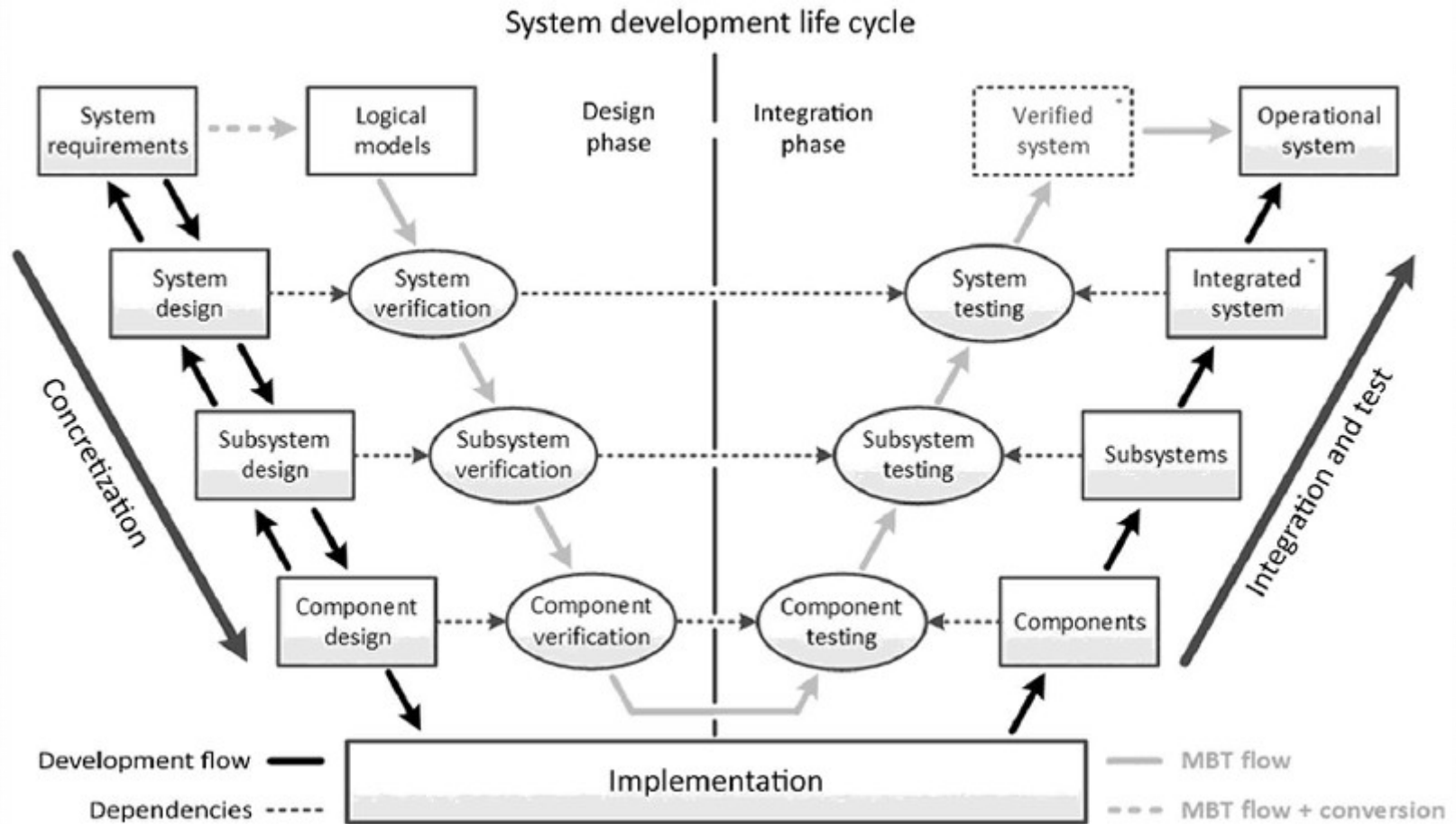
# SysML featurus

1. Simpler than UML (less diagrams)
2. Integrates hardware and software description
3. Possibility to intefrate other models i.e. output from ControlShell

# SysML a language for systems modelling



# V-Development life-cycle



\* Houbing Song, Danda B. Rawat, Sabina Jeschke, and Christian Brecher: Cyber-Physical Systems. Foundations, Principles, and Applications, Elsevier, Intelligent Data Centric Systems, 2017



# Application of formal methods



# Formal methods supporting modelling and verification. Qualitative approach

## **A. Describing behaviour and generating mainly LTS**

- (High Level) **Petri nets** (CPN Tools, UPAAL)
- Timed automata, Hybrid Automata** (UPAAL)
- Process Algebras** (LOTOS, LNT – CADP)

## **B. Logic based description – temporal logics: LTL, CTL\*, CTL, ...**

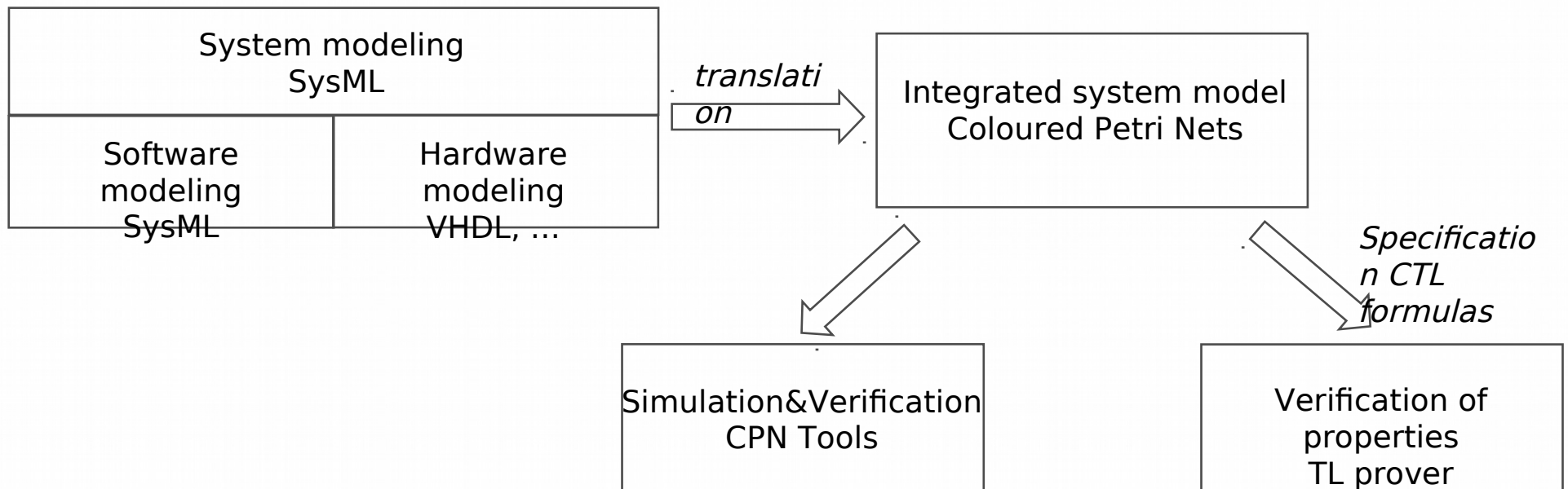
Proving using **Model Checkers** or **SAT Solvers**.



# Formal methods supporting modelling and verification. Quantitative approach

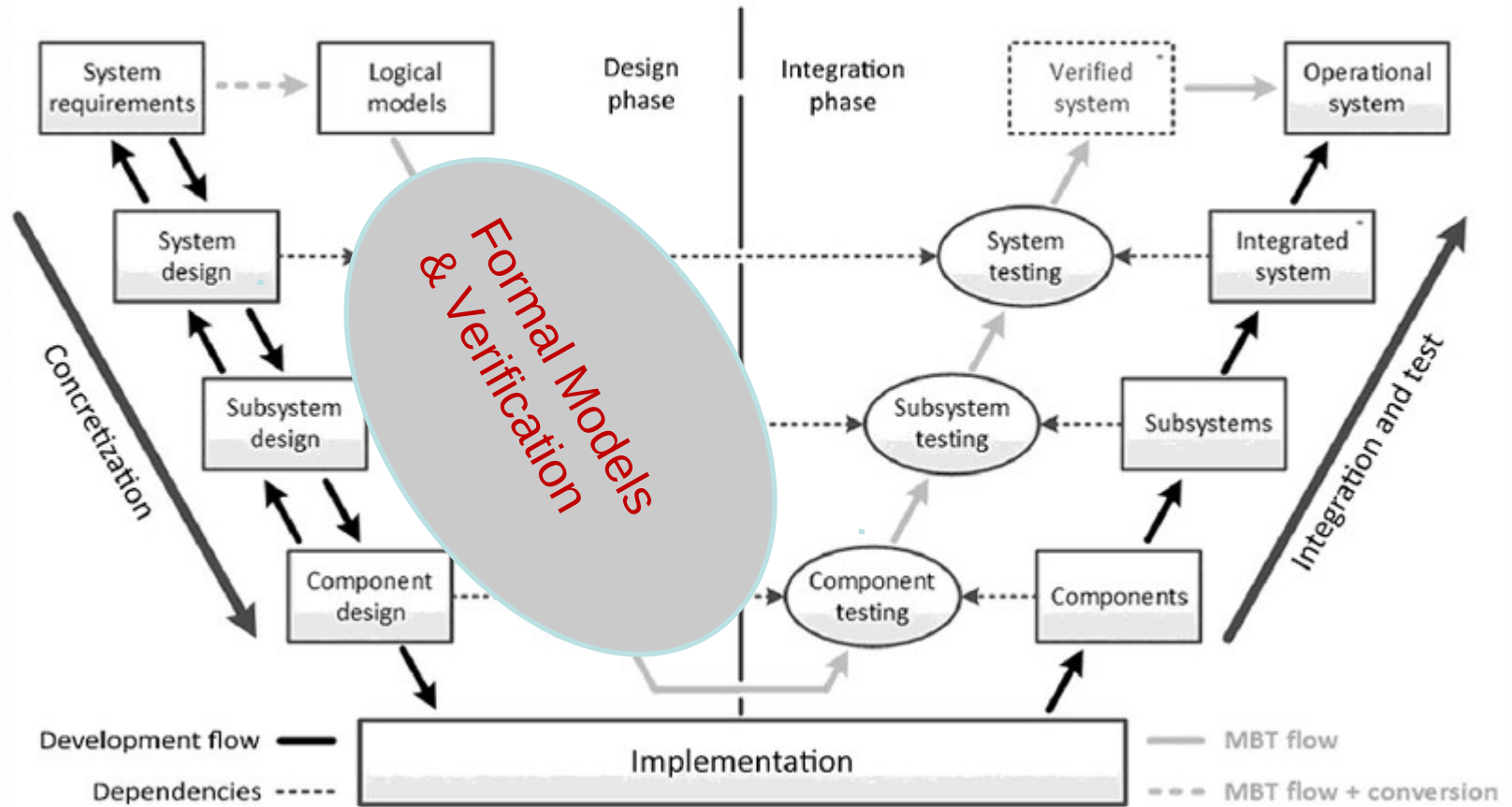
- A. Modelling of processes – Bayesian networks, Markov processes etc.
- B. Verification – probabilistic model checking (PRISM).

# Translation & verification



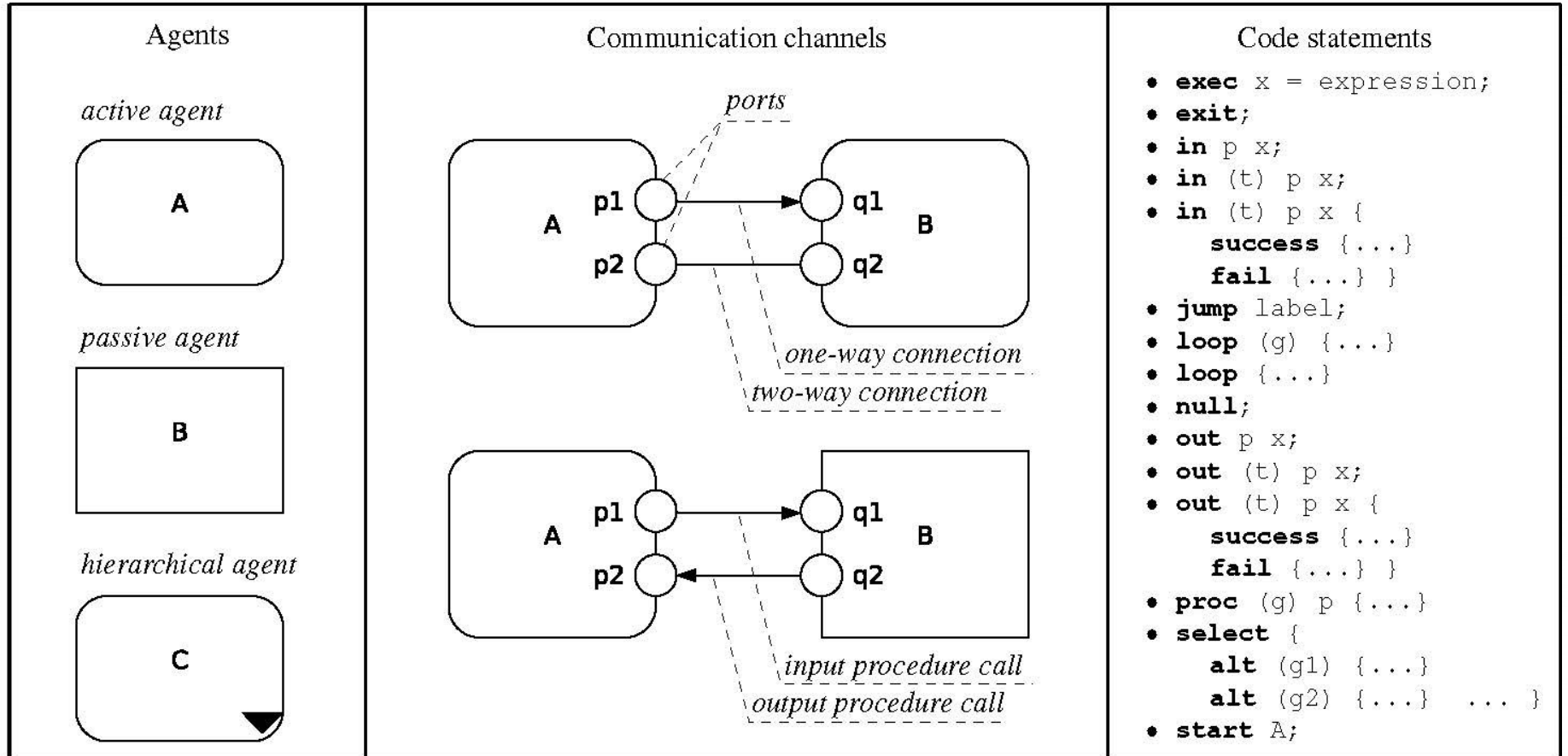
**Prototype of translator SysML artefacts → CPN**

### System development life cycle

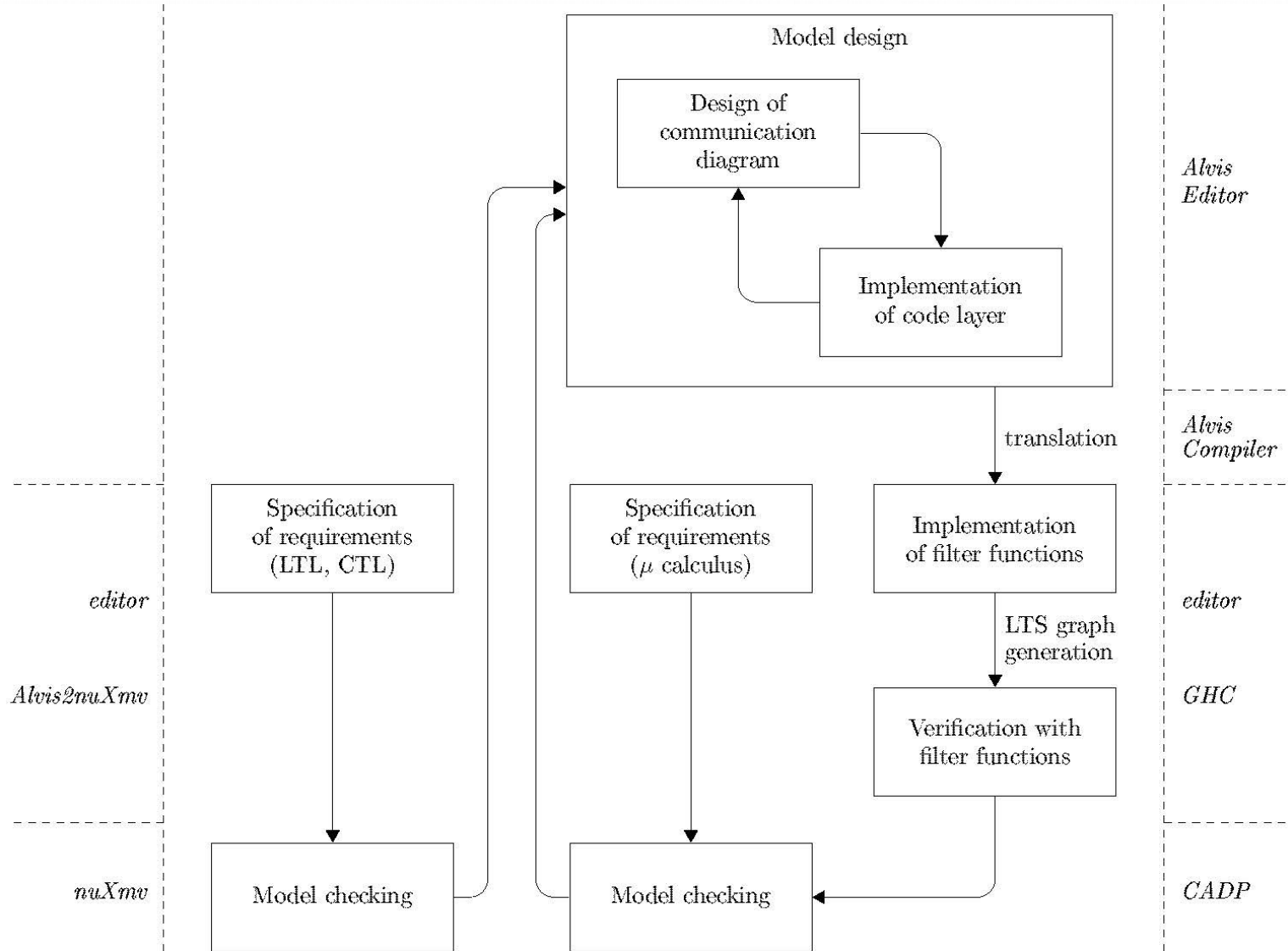




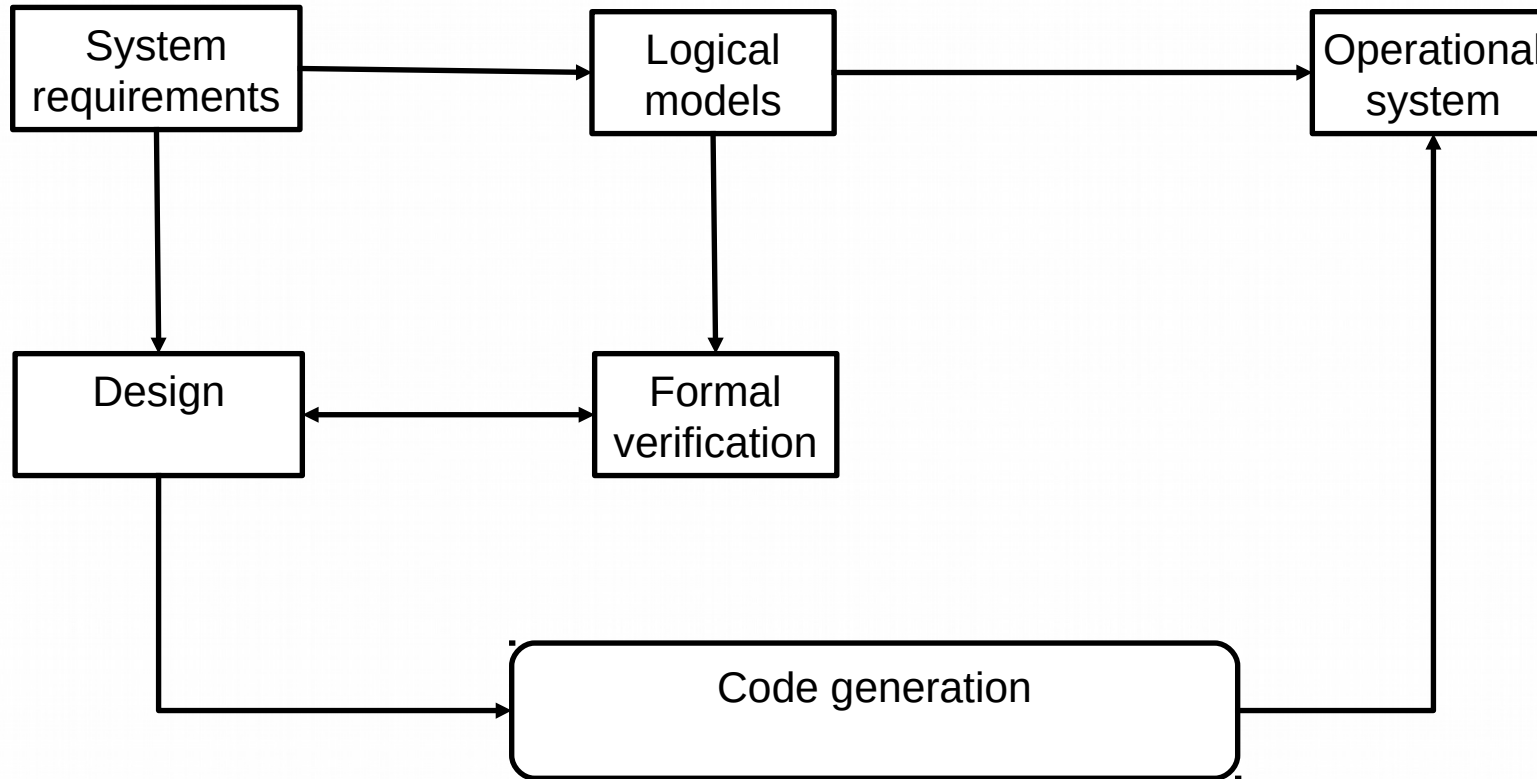
# Generation of software ALVIS

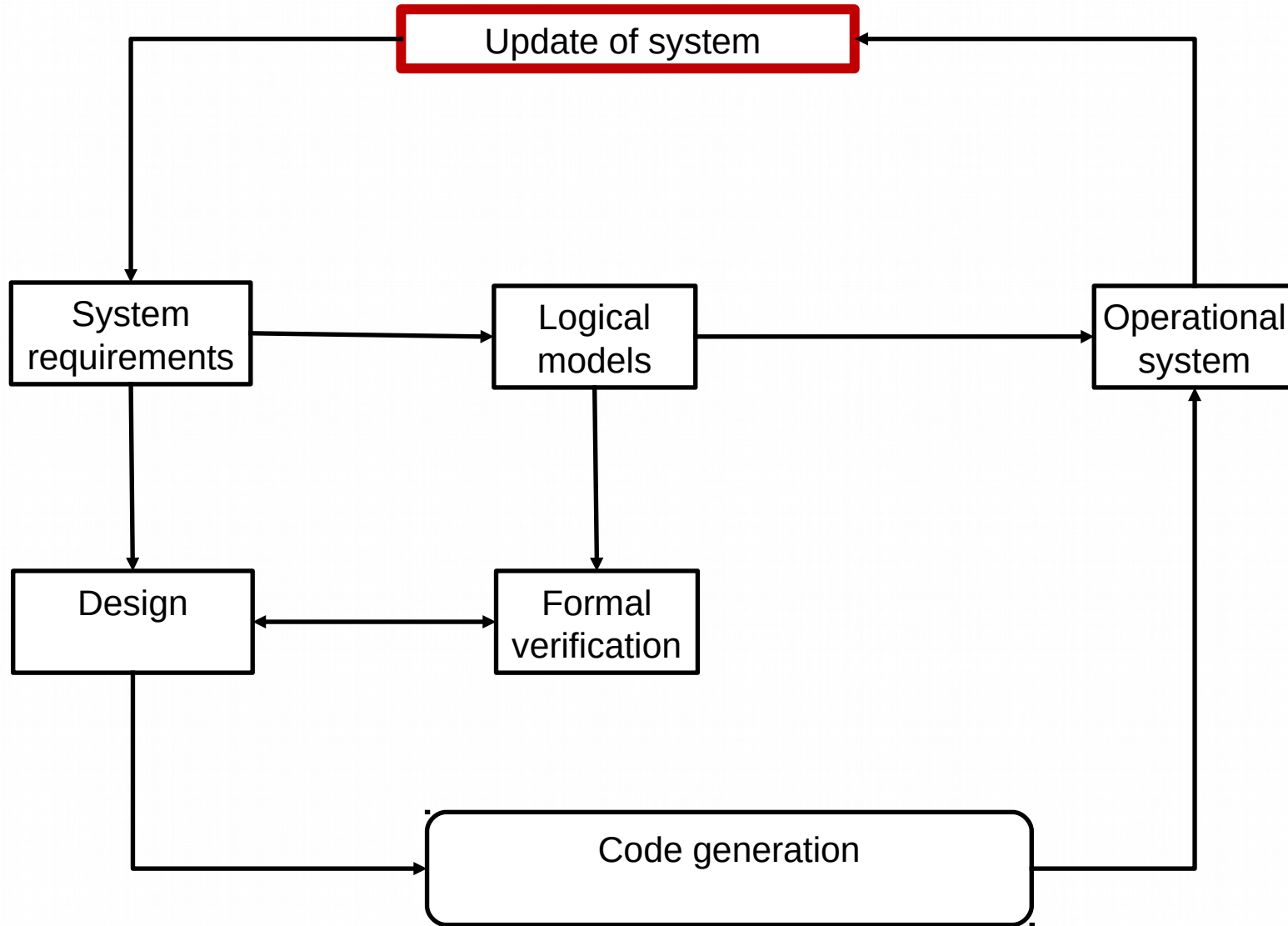


# Alvis process



# Is it possible to generate proper code on the basis of verified design?







## Application of DNN for generation of software

1. Reverse engineering – possible
2. Trainig DNN to generate program on a basis of specification and programming language?

## Conclusions

1. SysML seems to be sufficient for development of CPS systems. Some extensions may be applied by adding new type of blocks
2. Formal methods may improve software development esp. from integration & consistency point of view
3. Automation of design/code generation may speed-up development of CPS systems. Feedback in structure for construction of adaptive CPS?
4. Application of DNN for automatic generation of programs what are barriers?