Hasso Plattner Institut

Digital Engineering • Universität Potsdam

The 7th International Embedded Systems Symposium, IESS2022
3 -4 November 2022, Lippstadt, Germany

Towards Models for Smart Cyber-Physical Systems of Systems

Keynote at the 7th International Embedded Systems Symposium (IESS2022), 3 -4 November 2022, Lippstadt, Germany.

Holger Giese
System Analysis & Modeling Group,
Hasso Plattner Institute, University of Potsdam, Germany
holger.giese@hpi.uni-potsdam.de

Outline

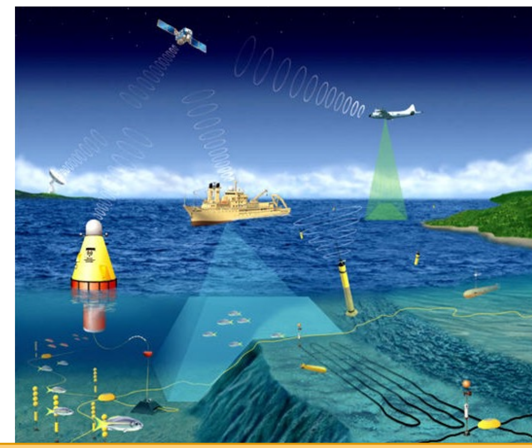
1. **Motivation**
2. **Being Smart**
3. **Smart & Learning**
4. **Conclusions & Outlook**

Outline

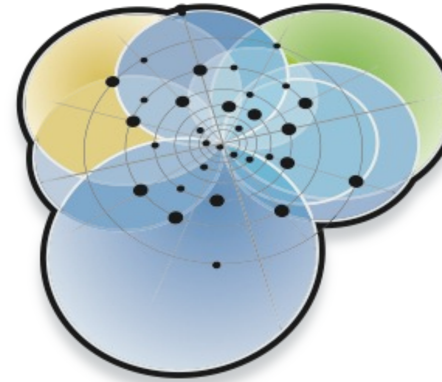
1. Motivation
The Future: You name it …

[Broy+2012]

Internet of Things



[Northrop+2006]



Ultra-Large-Scale Systems

(Networked)
Cyber-Phys

Smart Cyber-Physical Systems of Systems

E-Health

Smart Fact
E.g. Industry 4.0

Smart Home

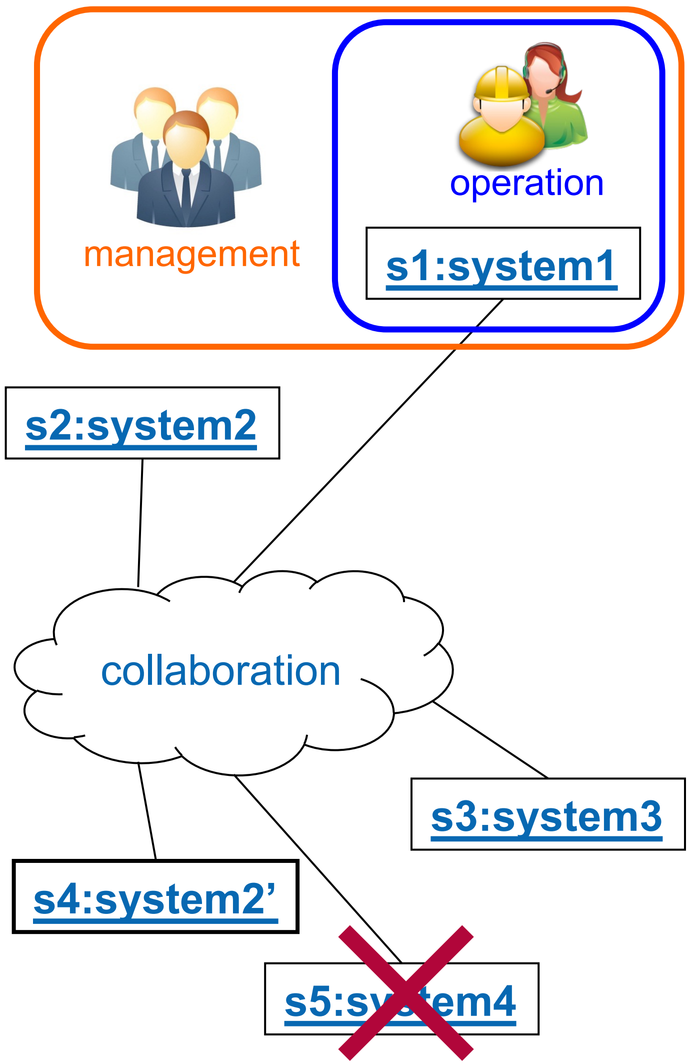
Smart Logistic

Ambient Assisted Living

Micro Grids

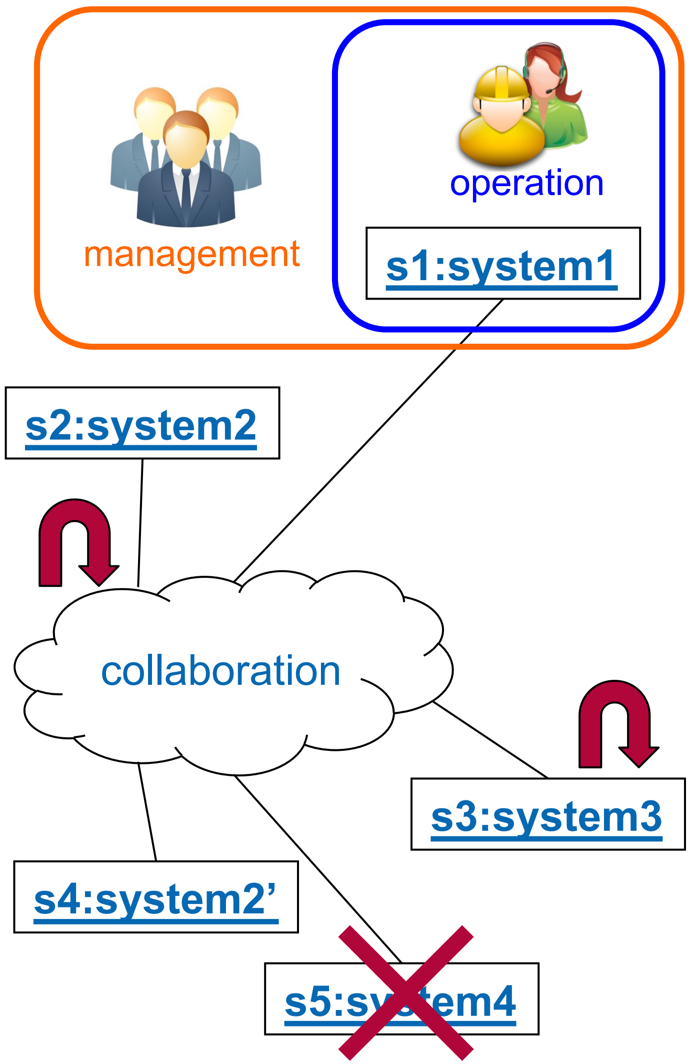
Smart City

A Selection of Critical Future Challenges (1/5)



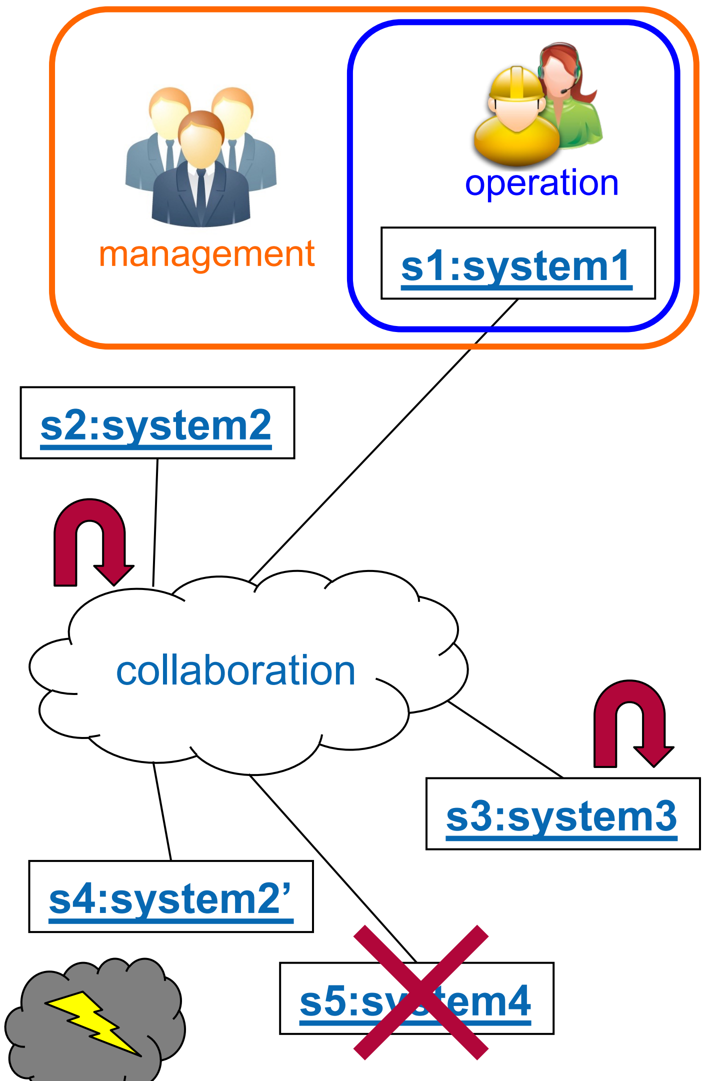
- **Operational** and **managerial independence**
  - operated independent from each other without global coordination
  - no centralized management decisions (possibly confliction decisions)
  - decoupled evolution and co-existence of different versions
- **Dynamic architecture** and **openness**
  - must be able to dynamically adapt/absorb structural deviations
  - subsystems may join or leave over time in a not pre-planned manner

A Selection of Critical Future Challenges (2/5)



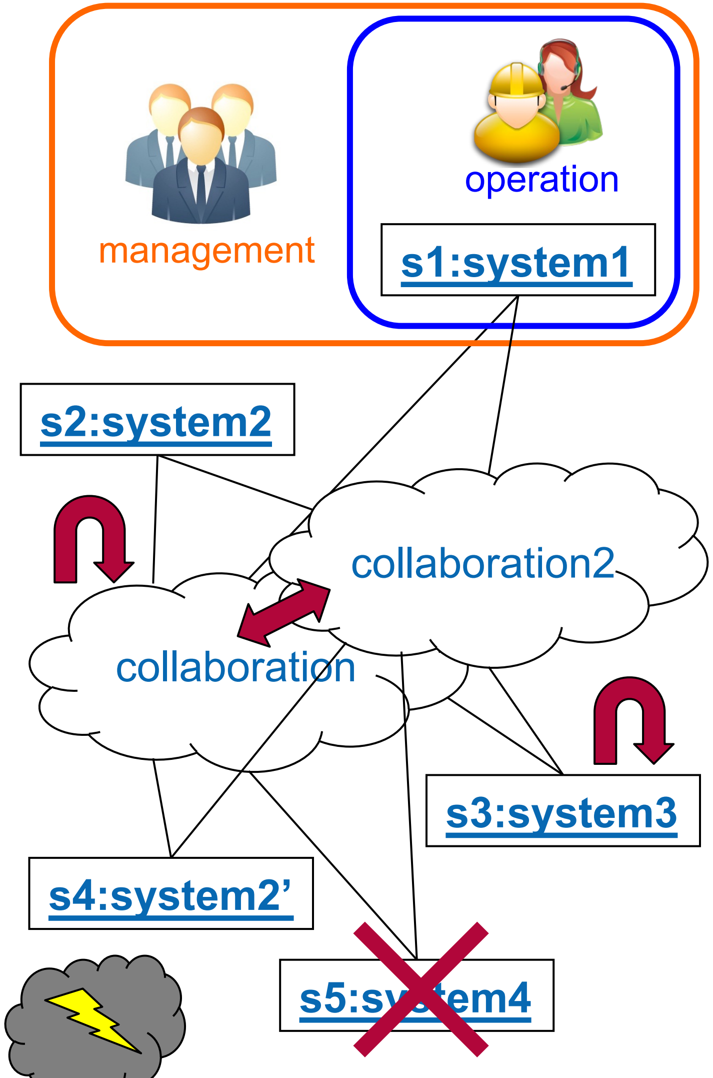
- **Operational** and **managerial independence**
  - operated independent from each other without global coordination
  - no centralized management decisions (possibly confliction decisions)
  - decoupled evolution and co-existence of different versions
- **Dynamic architecture** and **openness**
  - must be able to dynamically adapt/absorb structural deviations
  - subsystems may join or leave over time in a not pre-planned manner
- **Advanced adaptation**

A Selection of Critical Future Challenges (3/5)



- **Operational** and **managerial independence**
  - operated independent from each other without global coordination
  - no centralized management decisions (possibly confliction decisions)
  - decoupled evolution and co-existence of different versions
- **Dynamic architecture** and **openness**
  - must be able to dynamically adapt/absorb structural deviations
  - subsystems may join or leave over time in a not pre-planned manner
- **Advanced adaptation**
- **Resilience**

A Selection of Critical Future Challenges (4/5)

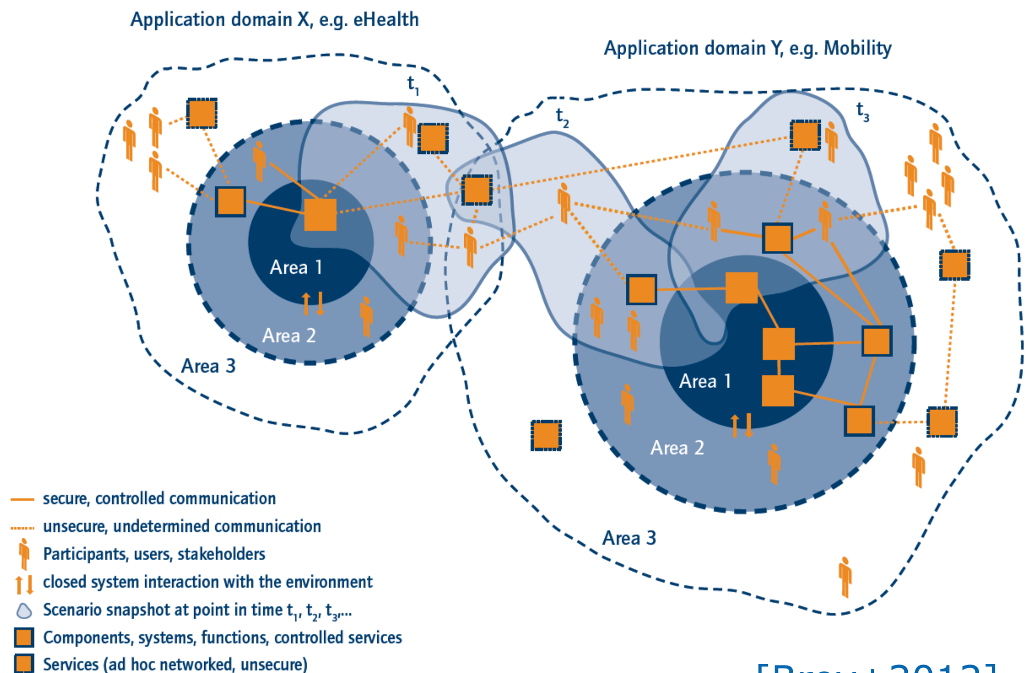


- **Operational** and **managerial independence**
  - operated independent from each other without global coordination
  - no centralized management decisions (possibly confliction decisions)
  - decoupled evolution and co-existence of different versions
- **Dynamic architecture** and **openness**
  - must be able to dynamically adapt/absorb structural deviations
  - subsystems may join or leave over time in a not pre-planned manner
- **Advanced adaptation**
- **Resilience**
- **Cross-Domain Integration**

Challenge: Cross-Domain Integration

HPI Hasso Plattner Institut

Example: A convoy of fully autonomous cars abandons the premium track in order to give way to an ambulance (intersection of CPS specific for **traffic** and **health care**)



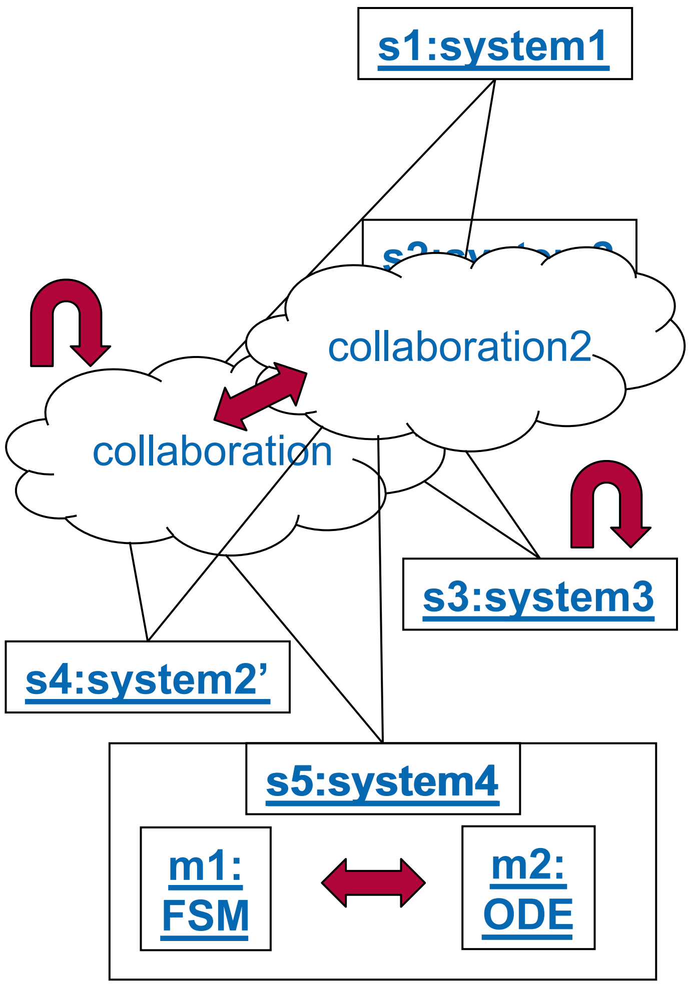
[Broy+2012]

CPS of different domains have to be connected:

- According to social and spatial network topologies, CPS operate across different nested spheres of uncertainty
- CPS dedicated to different domains have to to interact and coordinate.

Integration has to cover multiple domains and their **paradigms**

A Selection of Critical Future Challenges (5/5)

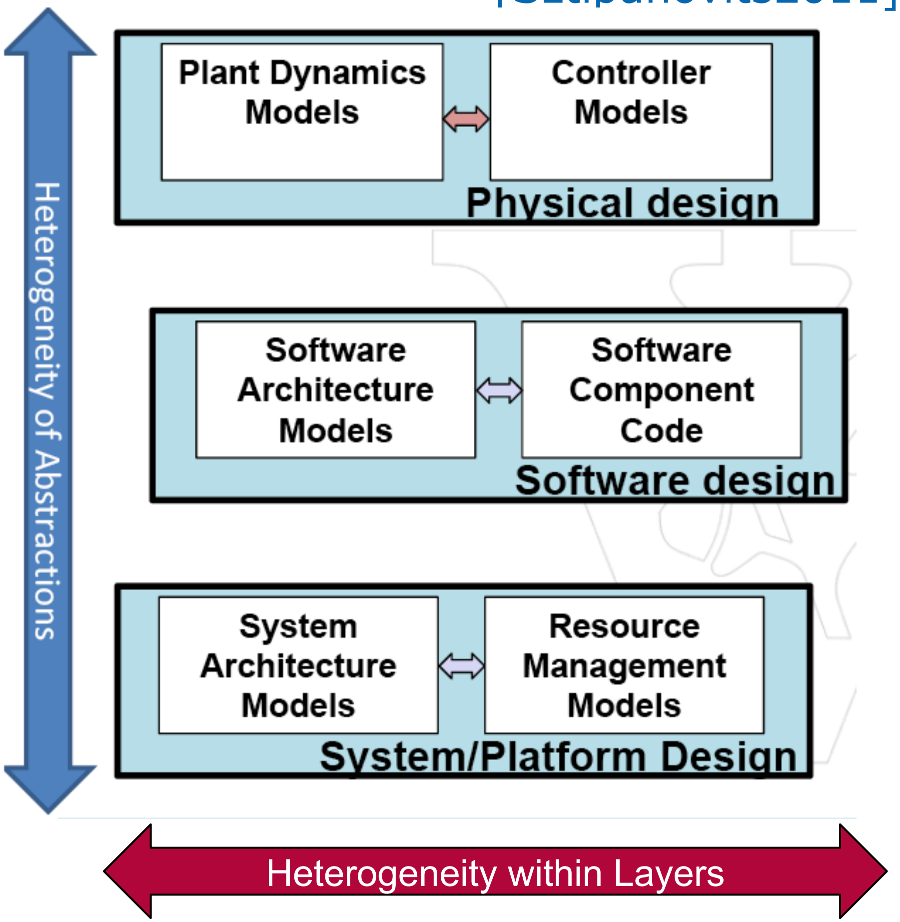


- **Operational** and **managerial independence**
  - operated independent from each other without global coordination
  - no centralized management decisions (possibly confliction decisions)
  - decoupled evolution and co-existence of different versions
- **Dynamic architecture** and **openness**
  - must be able to dynamically adapt/absorb structural deviations
  - subsystems may join or leave over time in a not pre-planned manner
- **Advanced adaptation**
- **Resilience**
- **Cross-Domain Integration**
- **Integrate Models of Computation**

Challenge: Integrate Models of Computation

- Problem to integrate models within one layer as different **models of computation** are employed
- Leaky abstractions are caused by lack of composability across system layers. Consequences:
  - intractable interactions
  - unpredictable system level behavior
  - full-system verification does not scale

[Sztipanovits2011]



Integration has to cover multiple layers and their paradigms

Outline

1. Motivation
2. **Being Smart**
3. Smart & Learning
4. Conclusions & Outlook

2. Being Smart: Adaptation Feedback Loop (MAPE-K)

Hasso Plattner Institut

Plan: what adaptation is "good" resp. "best"" for the made perceptions

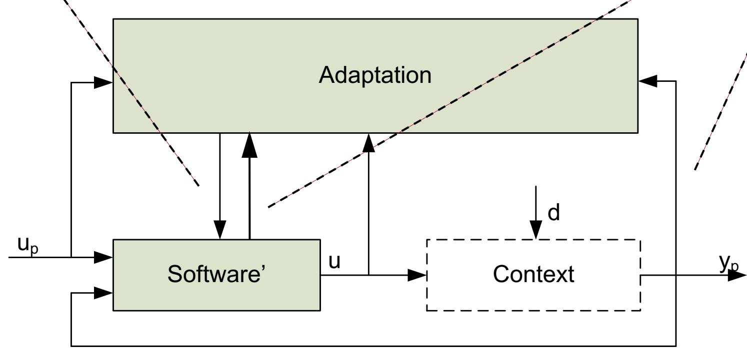
Analysis: is an adaptation "required"/beneficial for the made perceptions

Knowledge guiding the adaptation

- Simple: implicit
- Complex: partially explicit

Execute/Adapt: enact planned adaptation

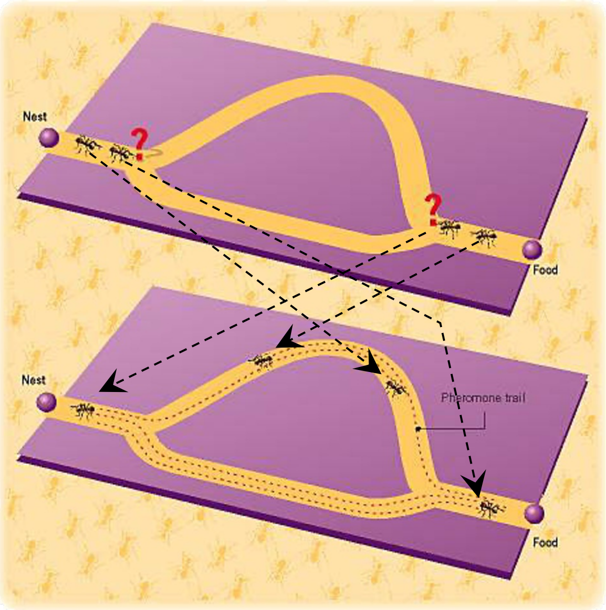
Monitor/Observe: how to interpret observations (**perception**)



Is Simple Self-Awareness Enough?

Simple Self-Awareness : Let’s have a look at Nature …

Ant colonies operate as a **superorganism** that combines information processing of many ants and their interaction with the environment at the physical level (using stigmergy as coordination mechanism).



Example:

- Asymmetric binary bridge experiment

Observations:

- Initially both options will be taken with the same probability.
- The concentration of the pheromones will increase faster on the shorter path.
- The higher concentration of pheromones on the shorter path will make it more likely that an ant choses this shorter one.
- Positive feedback will amplify this effect and thus finally the longer path will only be used seldom.

➜ Can our problems be solved by borrow ideas from nature?

Simple Self-Awareness : Let’s have a look at Nature ...

Another Example:

- “Ant Mill”

Observations:

- Such a behavior would be not acceptable for an engineered system even for **unexpected circumstances (rare events)**.
- If even “Nature” come up with designed solutions that fail (even evolution selected for ages), how could we envision to be more successful?
- But there is also a solution in nature:

reflect on self (**Complex Self-Awareness**)

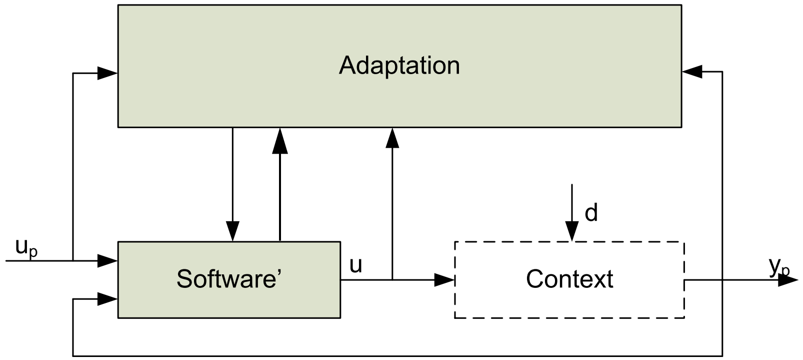
Complex Aware-ness: Runtime Models

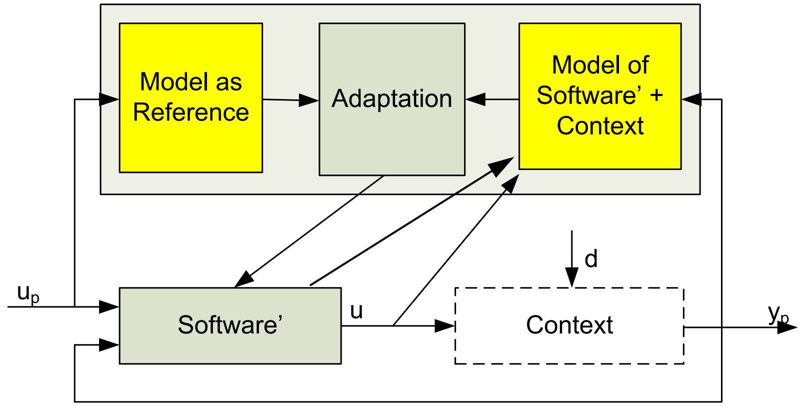
Simple Awareness "without" models:

- Still explicit or implicit **design-time models** are used guide adaptation processes

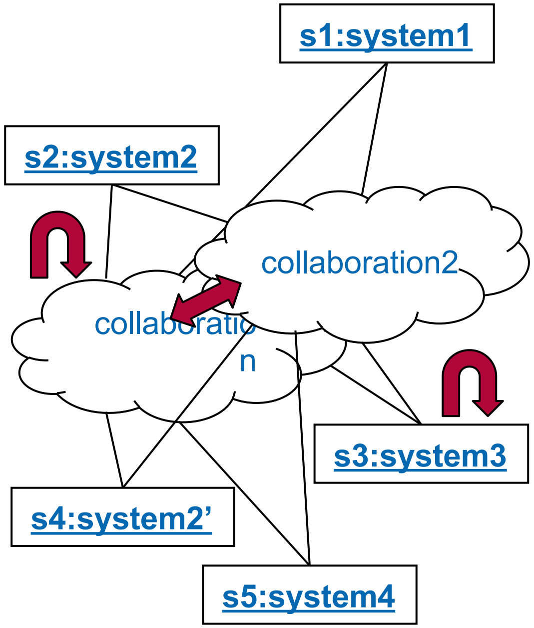
Complex Awareness with runtime models:

- Explicit **runtime models** are **learned** to guide adaptation processes and allow **reasoning**
- Model as reference can include goals
- Model of software + context capture changes
- **Limitation:** covers only changes captured by the runtime models (possibly **multiple**!); requires correct adjustment from observations
- **Option:** adaptation can also reason with these models (e.g., predict outcome of changes)





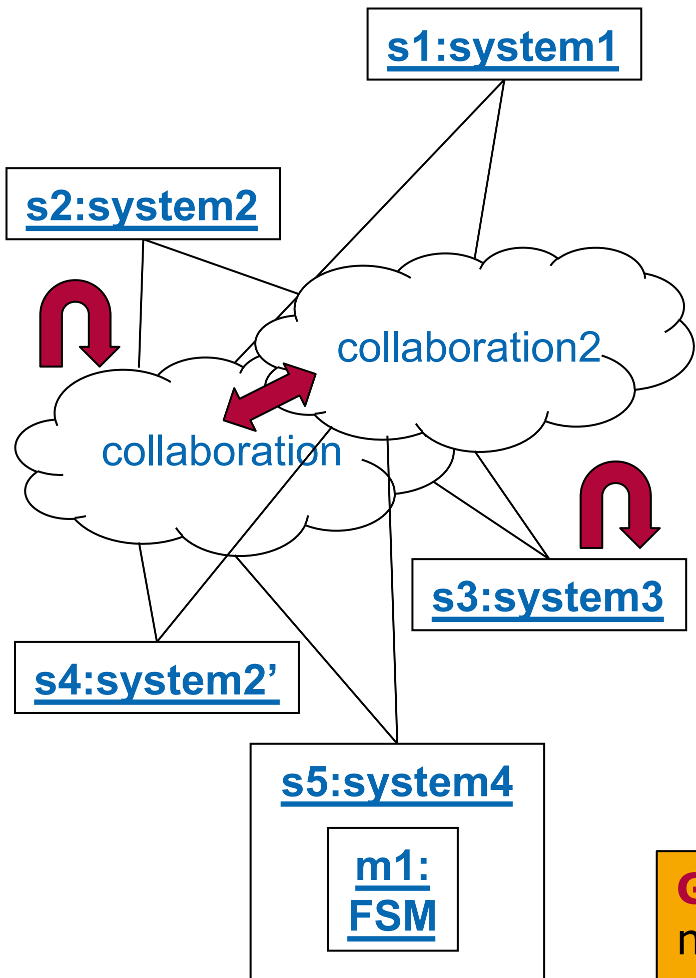
Self-Awareness & System of Systems



- **Self-Adaptive Systems:**
  - Make systems **self-aware**, context-aware, and requirements-aware using some form of **reflection**
  - Enable systems to adjust their structure/behavior accordingly
- **Self-Organization:**
  - The capability of a system of systems to organize their structure/behavior without a central control (emergent behavior)

- **Observations:**
  - a spectrum from centralized top-down self-adaptation to decentralized bottom-up self-organization with many intermediate forms exists
  - existing (formal) models and analysis approaches for CPS are no longer applicable as they do not cover reflection/adaptation (design, verification, ...)

Some Related Observations …



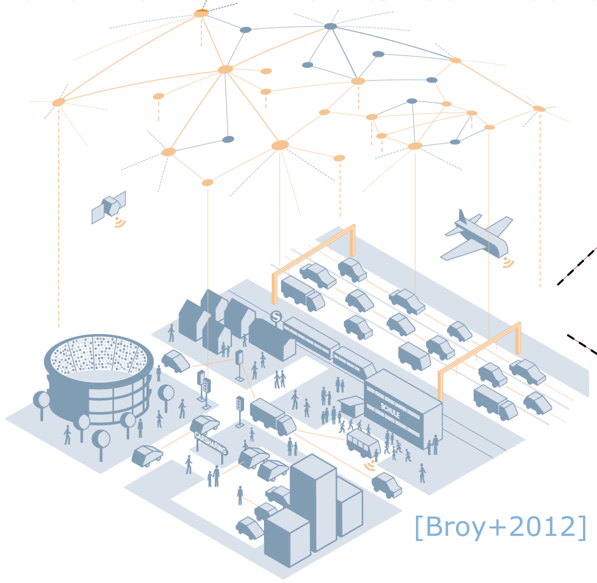
[Giese+2015]

- **Operational** and **managerial independence** as well as **dynamic architecture** and **openness** can be described by a **graph** of links between the systems that evolve
- **Self-Adaptive** and **Self-Organization** can be described by a **graph** of links between the components resp. systems that evolve/reconfigure and in case of **reflection** most models can be described by such a **graph** as well
- **Runtime Models** can be described by a dynamic **graph** of models and links between them

Graph transformation systems encoding models and their linking would allow to cover the challenges **…**

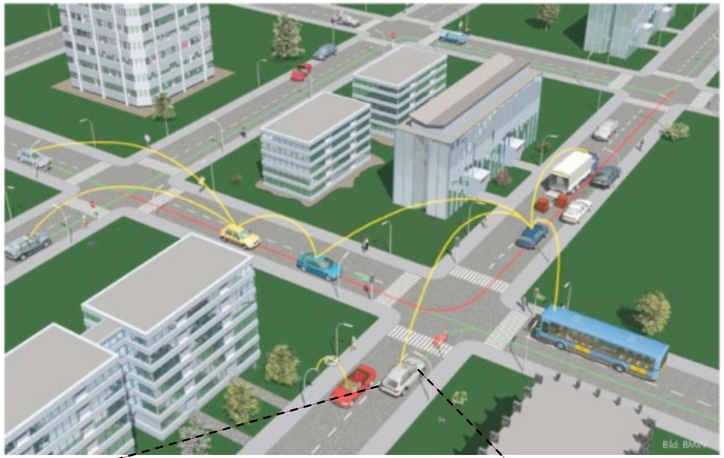
Complex Self-Awareness: Different Views/Levels …

Holistic Views on traffic/health care/… (systems of systems)



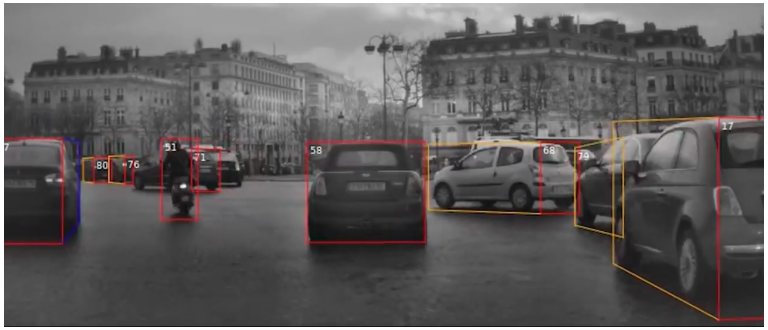
[Broy+2012]

View of a crossing (local system of systems)



http://www.innovisions-magazin.de/content/magazin/ausgabe/01..2006_8/themen/Ambient%20Intelligence_44/Gefahrenmeldungen+verbreiten+sich+wie+ein+Lauffeuer_546?PHPSESSID=bd6d23b3d56bea312d194a679dfd2830

View of a car (system)



https://www.youtube.com/watch?v=meTZKZp5QDY

Focus: The local system of systems and their systems with their runtime models

Complex Self-Awareness: A look at a complex example...

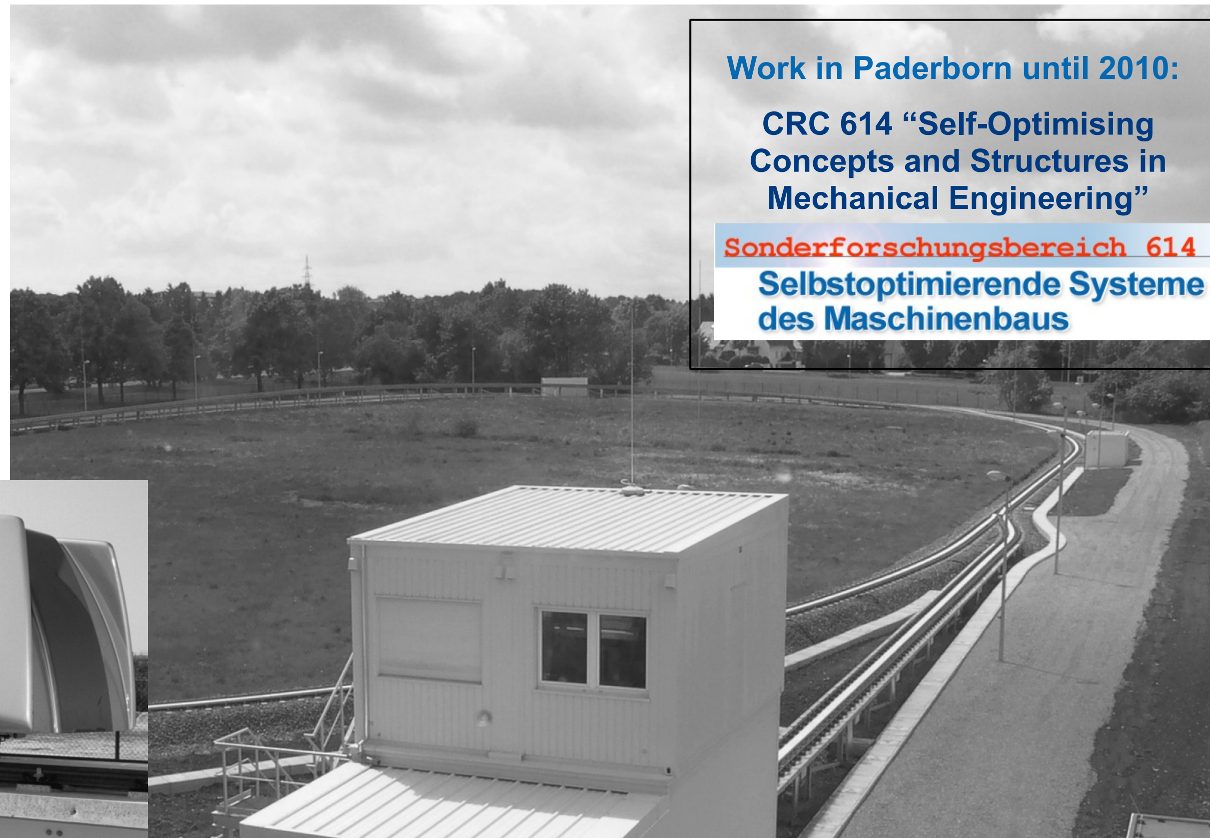
A system of **autonomous shuttles** that operate on demand and in a decentralized manner using a **wireless network**.

Smart CPS:

- Hard real-time
- Safety-critical
- Self-Optimization

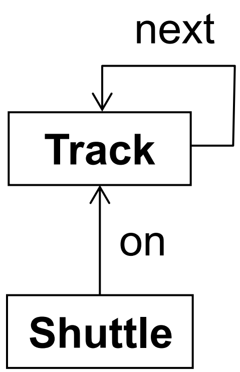
Needs:

- Optimized maneuvers, operation, and resource utilization (e.g., convoy)

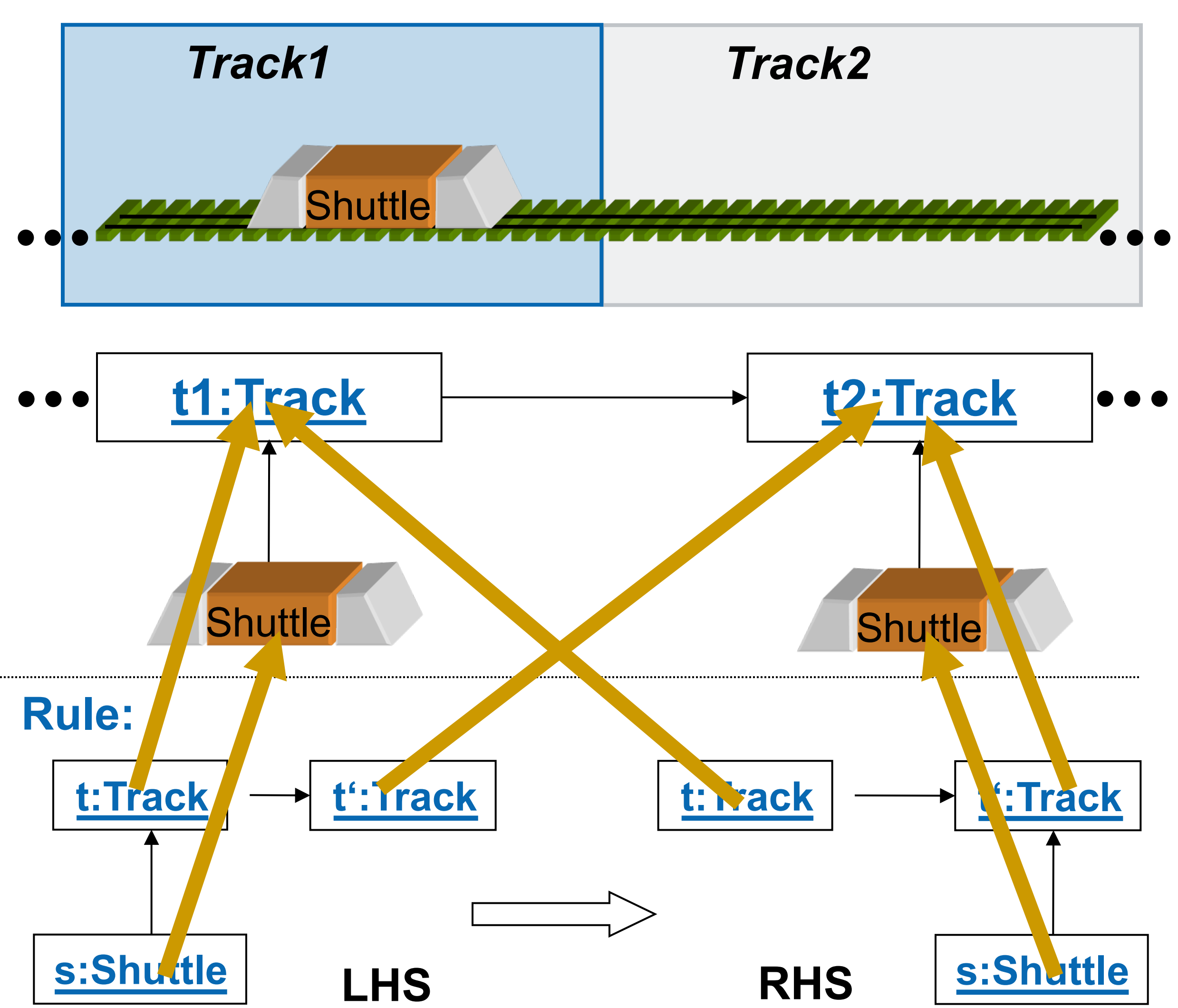


Graph Transformation Systems: Naïve Example

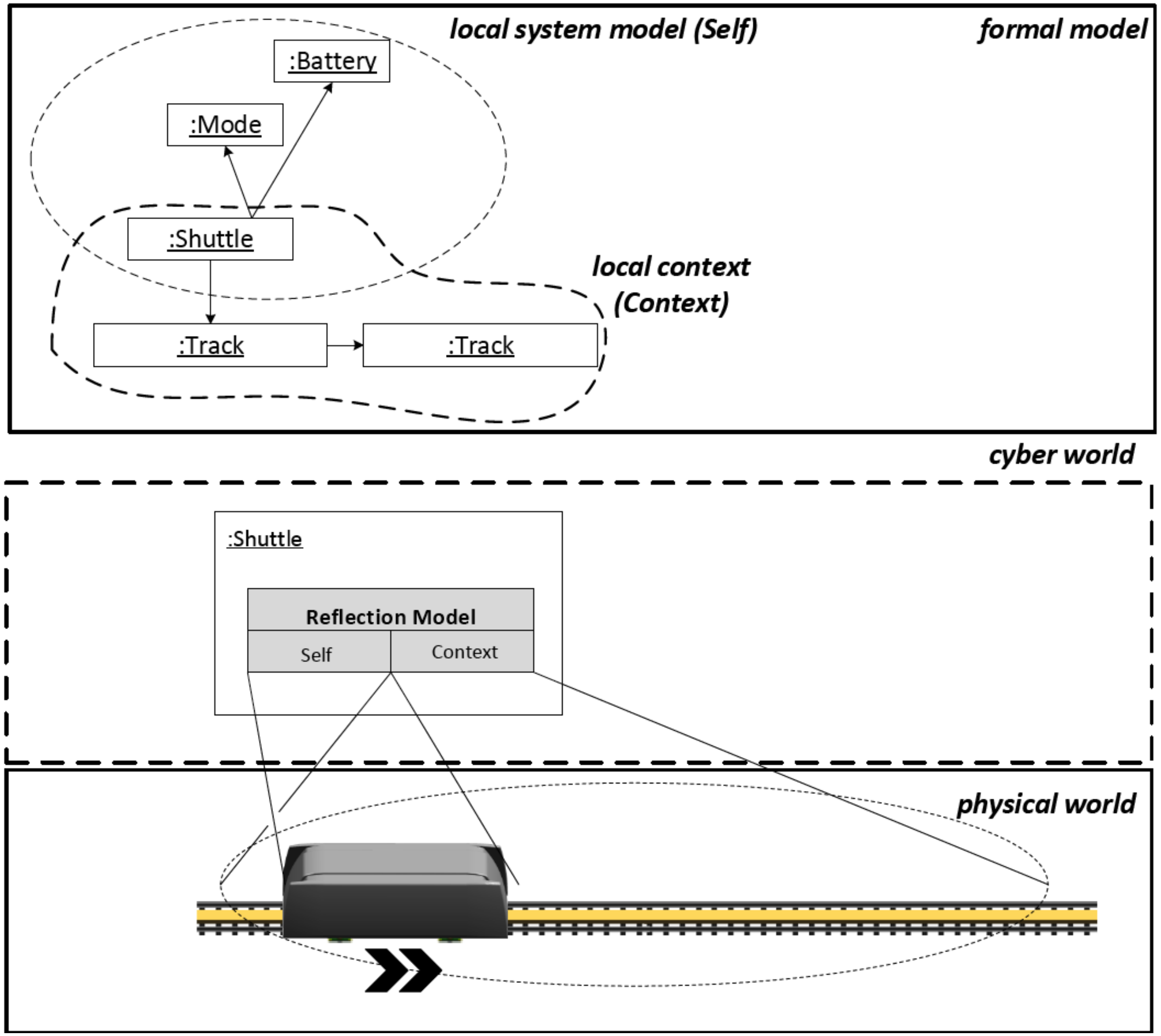
- Map the tracks
- Map the shuttles



- Map the movement to rules (movement equals dynamic structural adaptation on the abstract level)



Runtime Models & Idealized Perception



[Giese+2015]

SMARTSOS suggests:

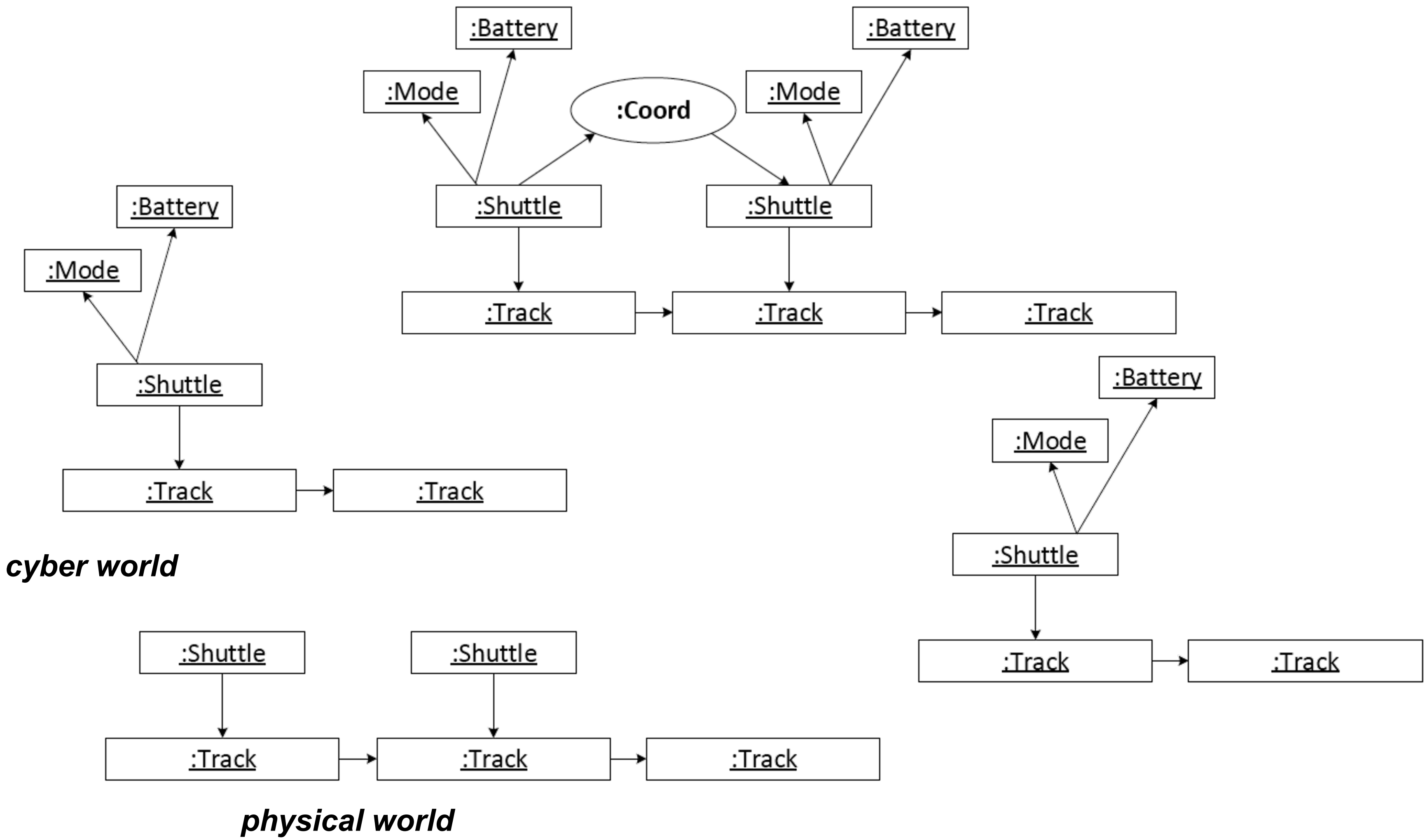
- use a **graph** of links between the systems, components, and internal represented data as well as
- use **graph transfor-mations** to capture possible changes

to model

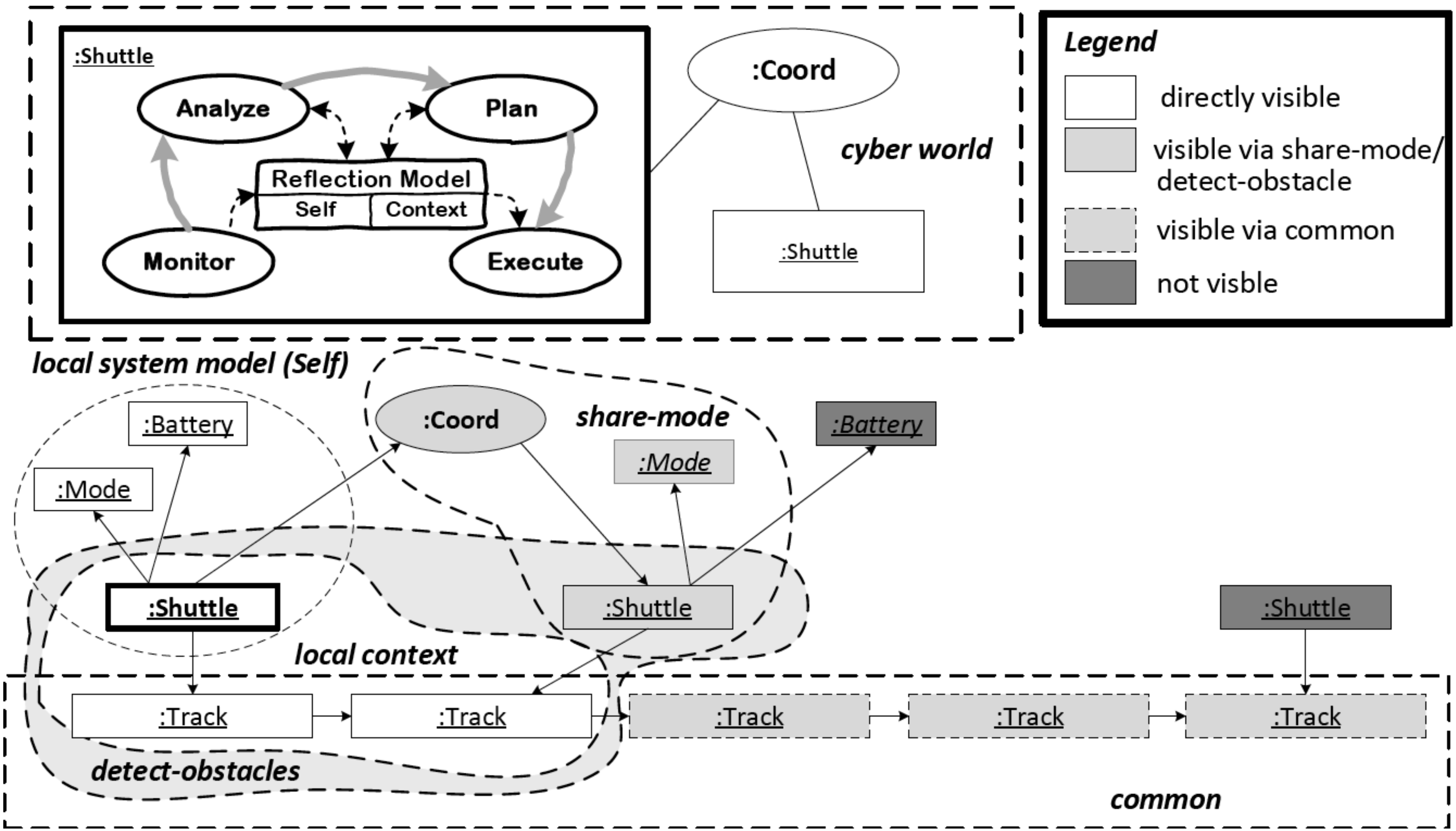
- **Service-Oriented Architecture**,
- **Self-Adaptive** and **Self-Organization**, and
- **Runtime Models**

Idealized Consistent Cyber & Physical World

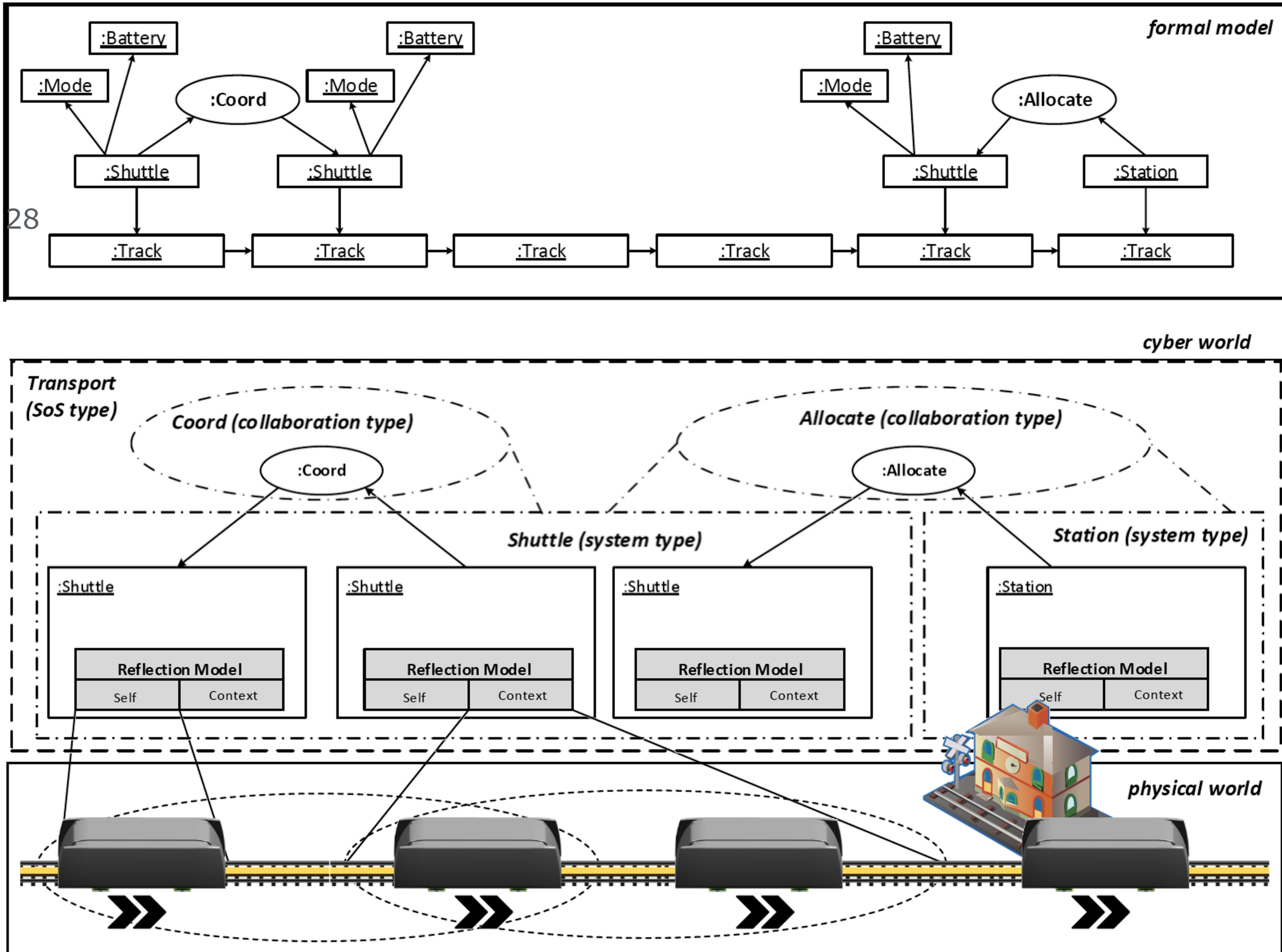
[Giese+2015]



Runtime Models & Exchanging Perceptions



[Giese+2015]



formal model
:Battery
:Battery
:Battery
:Mode
:Coord
:Mode
:Mode
:Allocate
:Shuttle
:Shuttle
:Shuttle
:Station
:Track
:Track
:Track
:Track
:Track
:Track
cyber world
Transport (SoS type)
Coord (collaboration type)
Allocate (collaboration type)
:Coord
:Allocate
Shuttle (system type)
Station (system type)
:Shuttle
:Shuttle
:Shuttle
:Station
Reflection Model
Self
Context
Reflection Model
Self
Context
Reflection Model
Self
Context
Reflection Model
Self
Context
physical world

Coverage of the Challenges for Models

Hasso Plattner Institut

Needs:

- **Operational** and **managerial independence**
- **Dynamic architecture** and **openness**
- **Scale** for local systems or networked resp. large-scale systems of systems
- **Integration** of the physical, cyber, (and social) dimension
- **Adaptation** at the system and system of system level
- **Decoupled evolution** and **co-existence of different versions**
- **Resilience** of the system of system

Model Characteristics:

- **Compositionality**
- **Dynamic structures**
- **Abstraction**
- **Hybrid behavior**
- **Non-deterministic**
- **Reflection for models**
- **Incremental extensions**
- **Probabilistic**

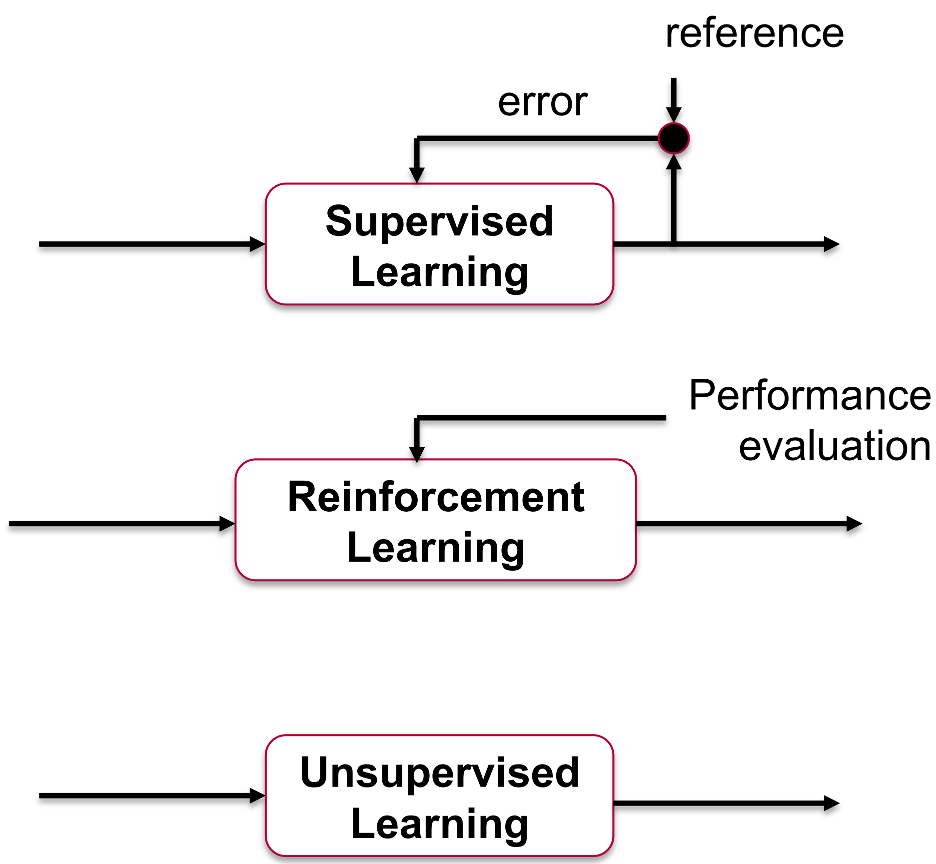
Our Work:

- **SMARTSOS (**employing Timed and Hybrid GTS [Giese+2015]**)**
- **Timed GTS ([**Becker&Giese2008]**)**
- **Hybrid GTS ([**Becker&Giese2012]**)**
- **MTGL for Timed GTS with History (**[Sakizloglou+2022|]**)**
- **Probabilistic timed GTS ([**Maximova2018]**)**
- **Probabilistic GTS ([**Krause&Giese2012]**)**

Outline

1. Motivation
2. Being Smart
3. **Smart & Learning**
4. Conclusions & Outlook

3. Smart & Learning: Learning vs. Training



Learning allows to adjust the behavior of systems:

- **Trained systems**:
  - learning only **offline**
  - **BUT:** additional surveillance must be **online**
- **Learning systems**:
  - Often initially **trained**
  - Steady improvement by learning **online**
  - additional surveillance needed?

Complex Self-Awareness & Learning

Hasso Plattner Institut

Train/Learn adaptation: …

Train/Learn goals: adjust goals according to success w.r.t. higher level goals

Variable goals: react to changes of the goals

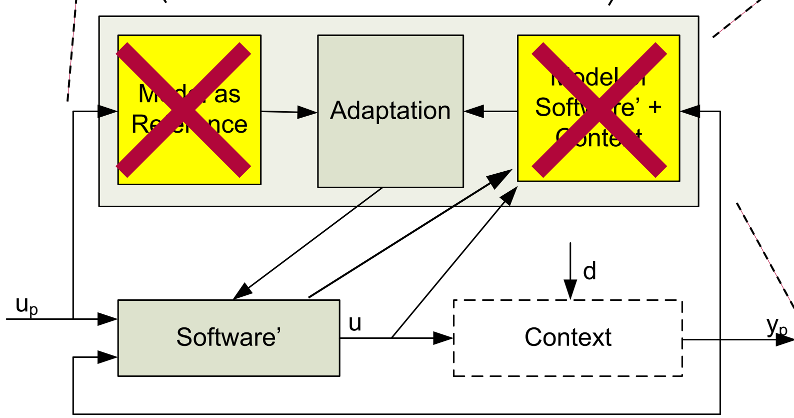
Static goals: no variation

Learn also runtime model concepts (unknown unknowns); runtime models evolve according to the perception of useful differences

Learn runtime models (known unknowns); parameters, elements, and relations of runtime models are **learned** according to the perception

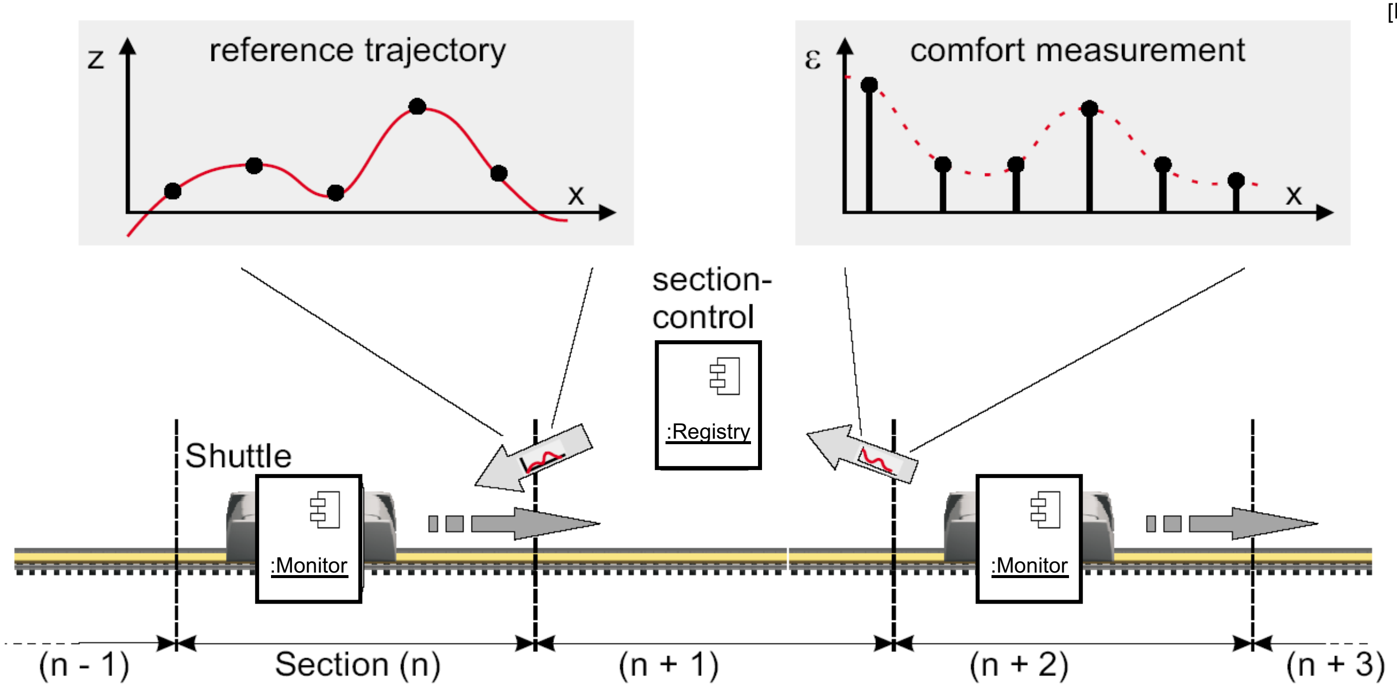
No learning: react to perceptions directly; avoid explicit model at run-time

Train/Learn perception: how to interpret observations



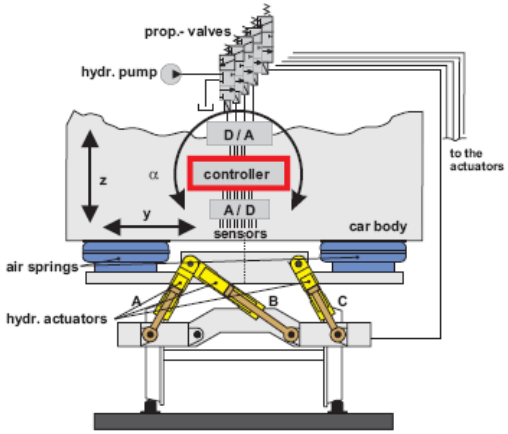
Complex Self-Awareness & Learn Context (1/3)

[Burmester+2008]



- **Server (Registry of the section control; not global!):**
  - Offers track profile (distributed learning of a runtime model of the track)
- **Client (Monitor of the shuttle):**
  - Applies track profile (local learning of a runtime model of the shuttle and planning an adaptation in form of an optimal trajectory)
  - Must handle cases where the service is available or not

Complex Self-Awareness & Learn Context (2/3)



Suspension/tilt module

- air springs (filter for higher frequencies)
- active suspension system (lower frequencies)

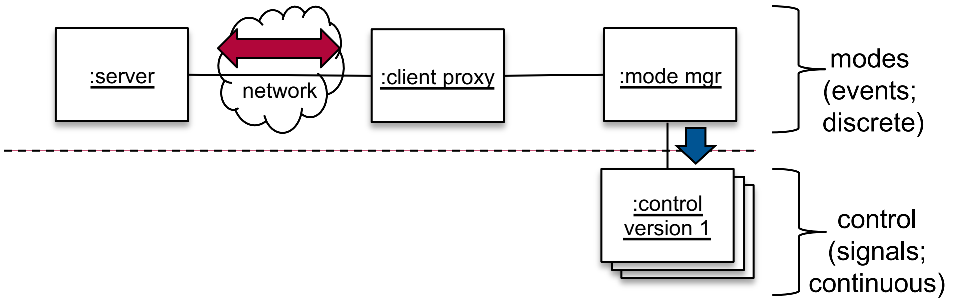
We consider three different control strategies:

(1) **robust controller:** track as reference point; damping the relative movement
⇨ only achieves moderate damping.

(2) **absolute controller** uses a virtual skyhook in order to ensure the absolute acceleration of the shuttle body is minimized
⇨ comfort usually maximized; problematic on inclines

(3) **reference controller**: Instead of virtual skyhook, the real track is used as reference
⇨ highest comfort; requires data about the track

Scheme:



Client proxy:

- Find local responsible registry
- register at the local registry (requestInfo)
- Receive data from the registry (sendInfo)
- Manage cases where the data is available or not (outside the proxy)
- Send data to the registry (experience)
- **PLUS:** detect invalid runtime model!

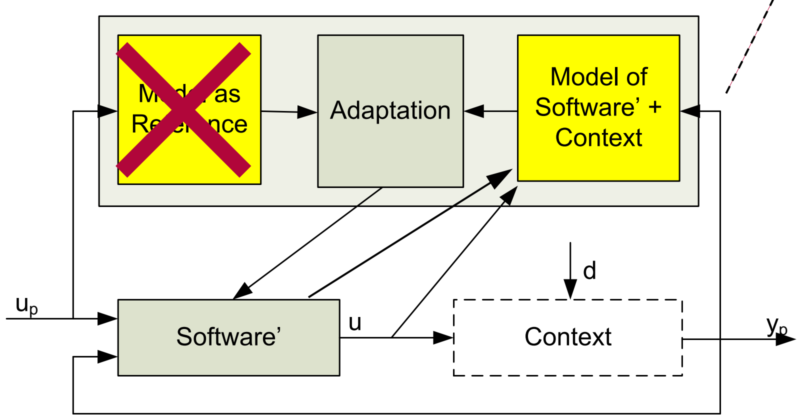
[Burmester+2008]

Complex Self-Awareness & Learn Context (3/3)

Distributed learning runtime models (known unknowns); parameters, ~~elements, and relations~~ of runtime models are **learned** according to the perception of other agents

Static goals: no variation

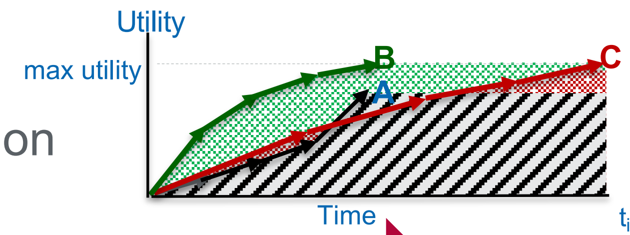
Learning runtime models (known unknowns); parameters, ~~elements, and relations~~ of runtime models are **learned** according to the perception



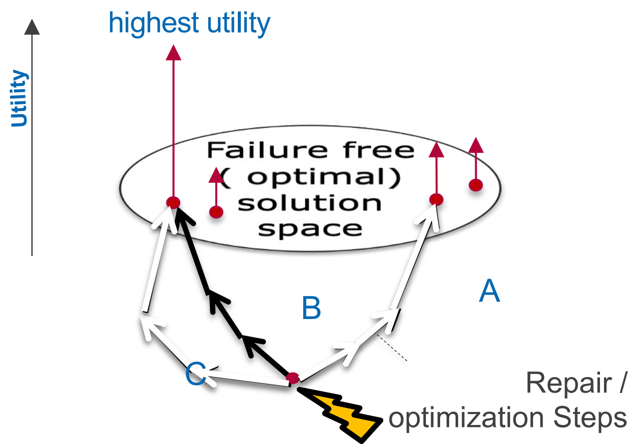
OBSERVATION: There is no guarantee that the runtime models are not invalid due to fact that they always rely on potentially erroneous or outdated measurements/ perceptions → **detection + backup strategy always necessary**

Complex Self-Awareness & Train Goals (1/4)

[Ghahremani+2017]
[Ghahremani+2018]



Self-adaptive systems that are rule-based
Architecture-based self-healing and self-optimization



	Optimization -based (C)	Rule -based (A)	Utility -driven (B)
Optimal order of repairs	✓	✗	✓
Scalable	✗	✓	✓
Maximum Utility	✓	✗	✓
Expressiveness	+	–	+/–

Required: Function computing the impact on the utility for each possible rule application

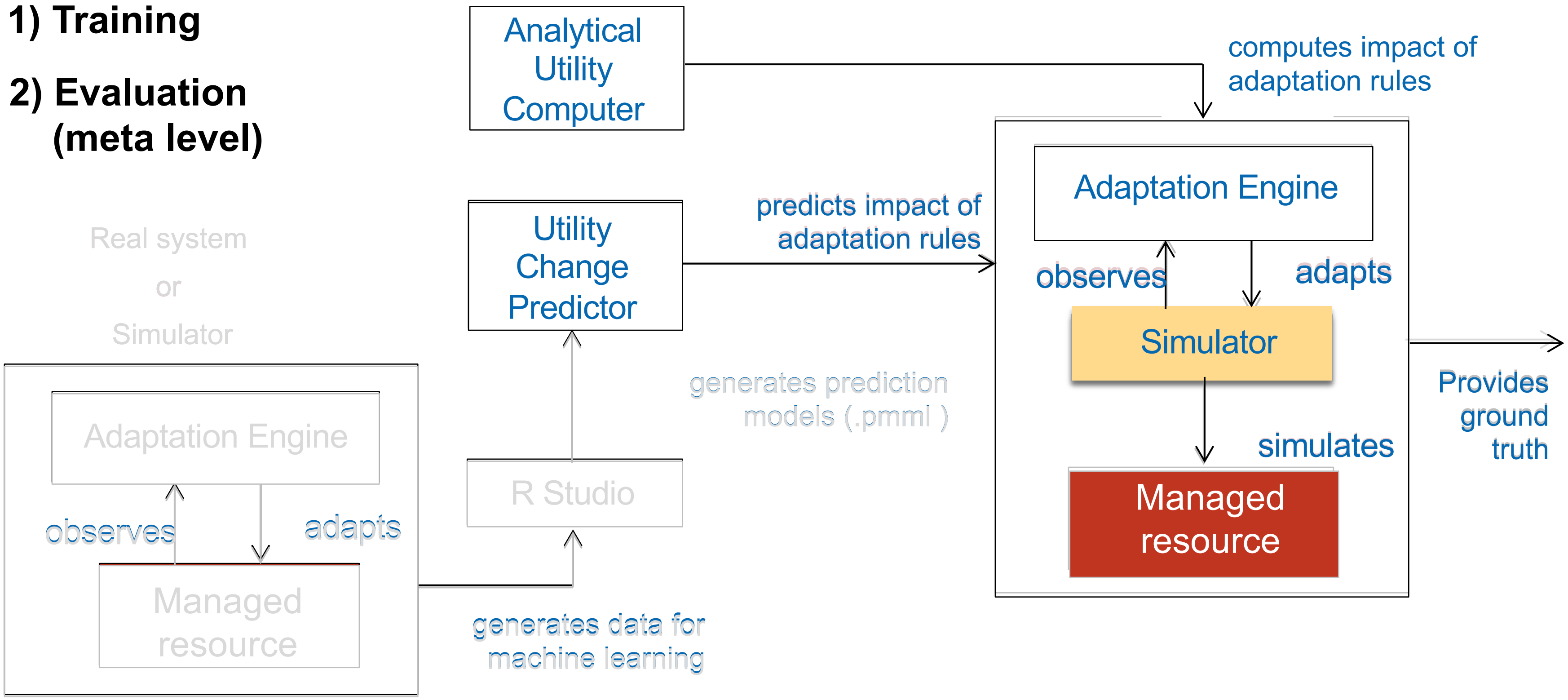
Open Question: Can we learn these functions offline (training)?

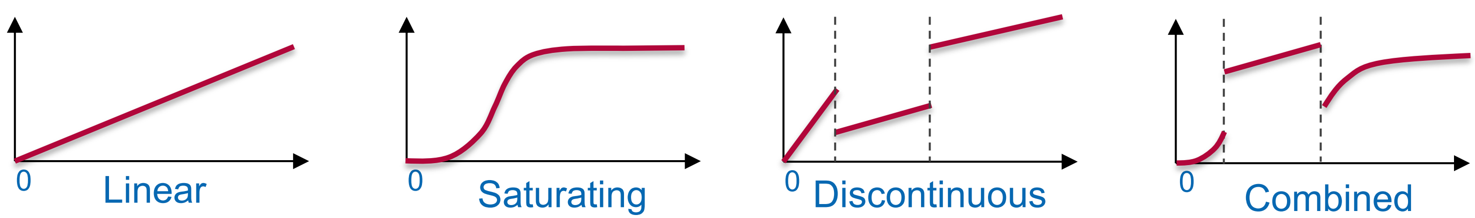
Complex Self-Awareness & Train Goals (2/4)

[Ghahremani+2018]

1) Training

2) Evaluation (meta level)



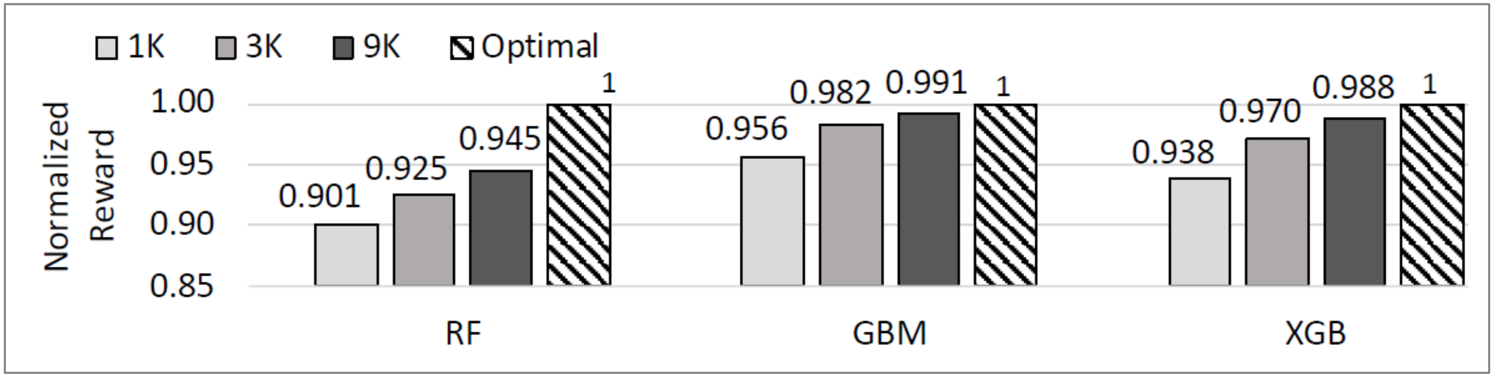


Complex Self-Awareness & Train Goals (3/4)

[Ghahremani+2018]

RQ: Does the performance approximate the analytic-defined optimum?

YES



Normalized rewards across prediction models for the **combined** variant

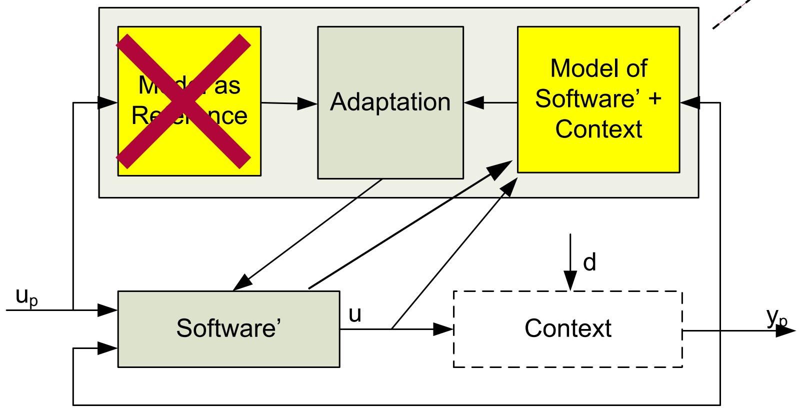
$$\text{Normalized Reward (mod)} = \frac{Reward\ (mod) - Reward(Baseline)}{Reward\ (Optimal) - Reward(Baseline)}$$

Complex Self-Awareness & Train Goals (4/4)

Train goals: adjust goals according to success w.r.t. higher level goals

PROBLEM: There is no guarantee that the trained goals are valid due to fact that they always rely on potentially erroneous or outdated measurements/perceptions
→ **optimality is not guaranteed**

Learn runtime models (known unknowns); parameters, elements, and relations of runtime models are **learned** according to the perception



Outline

1. Motivation
2. Being Smart
3. Smart & Learning
4. **Conclusions & Outlook**

4. Conclusions & Outlook

Graph transformation systems encoding models and their linking would allow to cover the challenges **…**

- **Operational** and **managerial independence** as well as **dynamic architecture** and **openness** can be described
- Runtime models and via collaborations shared runtime models enabled **Self-Adaptation** and **Self-Organization** of the systems and system of system
- **Limitations:**
  - Model is a rather strong **idealization**
  - Analysis relies on the **validity/trustworthiness** of the models
    - Development-time models may become invalid over time
    - Runtime models for self-awareness may become invalid

Outlook (1/5)
Modeling

Needs:

- **Operational** and **managerial independence**
- **Dynamic architecture** and **openness**
- **Scale** for local systems or networked resp. large-scale systems of systems
- **Integration** of the physical, cyber, (and social) dimension
- **Adaptation** at the system and system of system level
- **Decoupled evolution** and **co-existence of different versions**
- **Resilience** of the system of system

Model Characteristics:

- **Compositionality**
- **Dynamic structures**
- **Abstraction**
- **Hybrid behavior**
- **Non-deterministic**
- **Reflection for models**
- **Incremental extensions**
- **Probabilistic**

Our Work:

- **SMARTSOS (**employing Timed and Hybrid GTS [Giese+2015]**)**
- **Timed GTS** **(**[Becker&Giese2008]**)**
- **Hybrid GTS** **(**[Becker&Giese2012]**)**
- **MTGL for Timed GTS with History (**[Sakizloglou+2022]**)**
- **Probabilistic GTS** **(**[Krause&Giese2012]**)**

Probabilistic timed GTS ([Maximova2018]**)**

?

BUT: We would need as foundation formalisms that supports all required characteristics at once!

Outlook (2/5)

Analysis, Implementation, …

Hasso Plattner Institut

AND: cover the learning …

Needs:

- **Operational** and **managerial independence**
- **Dynamic architecture** and **openness**
- **Scale** for local systems or networked resp. large-scale systems of systems
- **Integration** of the physical, cyber, (and social) dimension
- **Adaptation** at the system and system of system level
- **Decoupled evolution** and **co-existence of different versions**
- **Resilience** of the system of system

Model Characteristics:

- **Compositionality**
- **Dynamic structures**
- **Abstraction**
- **Hybrid behavior**
- **Non-deterministic**
- **Reflection for models**
- **Incremental extensions**
- **Probabilistic**

SMARTSOS

State-of-the-Art & our Work:

- **Checking Inductive Invariants for GTS** ([Becker+2006]), **Timed GTS** ([Becker&Giese2008]), **and Hybrid GTS** ([Becker&Giese2012]) **and Checking k-Inductive Invariants for GTS** ([Dyck&Giese2017]) **resp. Attributed GTS** ([Schneider+2020])

  Only state properties!
- **Simulation for Probabilistic timed GTS** ([Zöllner+2021])

  Incomplete!
- **Model Checking Hybrid Systems**

  Only sequence properties for finite discrete state systems with rather bad scalability!
- **Model Checking Probabilistic GTS** ([Krause&Giese2012])
- **Probabilistic timed GTS: Model Checking** ([Maximova+2018]), **Compositional Model Checking** ([Maximova+2021])

  Only very restricted probabilistic sequence properties …

BUT: We have to assure resilience for complex se

AND: The implementation must preserve the assurances (problem: idealization).

Outlook (3/5)

Any approaches supporting Operational and Managerial Independence, Dynamic Architecture and Openness, Self-Adaptive / Self-Organization, and Runtime Models with evolving structures and Training/Learning will face tough challenges:

- **Challenges:**
  - Models have to capture many characteristics (e.g., dynamic structure, hybrid) usually tackled using different paradigms
  - Even in case of strong **idealization** often relevant properties can not be analyzed
  - Analysis relies on the **validity/trustworthiness** of the models
    - Development-time models may become invalid over time
    - Runtime models for self-awareness may become invalid

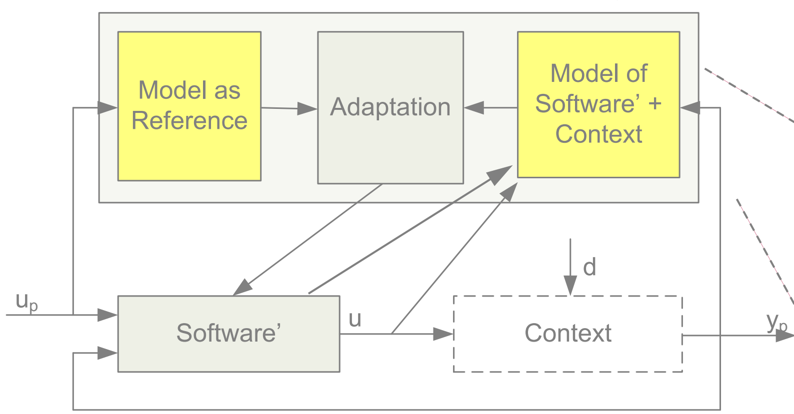
Outlook (4/5)

Runtime models for self-awareness **may** become invalid

Train/Learn goals: adjust goals according to success w.r.t. higher level goals

Train/Learn adaptation: …

Learn also runtime model concepts (unknown unknowns); runtime models evolve according to the perception of useful differences



Learn runtime models (known unknowns); parameters, elements, and relations of runtime models are **learned** according to the perception

Train/Learn perception: how to interpret observations

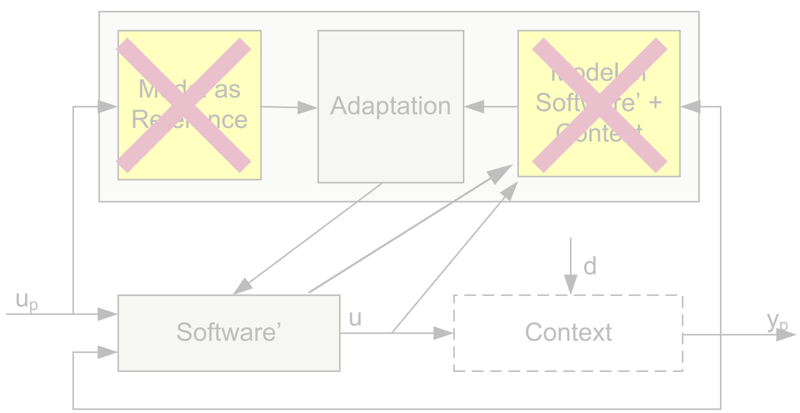
Correctness of the training/learning is crucial, if we have no **robust backup**, and guarantees are required

Outlook (5/5)

Runtime models for self-awareness **may** become invalid

BUT: Development-time models **will** become invalid over time

Static goals:
no variation



No learning: react to perceptions directly; avoid explicit model at run-time

Avoiding learning is also not safe (in the long run) if we do not have a **robust solution** that covers all possible future changes.

Bibliography (1/4)

[Broman+2012] David Broman, Edward A. Lee, Stavros Tripakis and Martin Torngren. Viewpoints, Formalisms, Languages, and Tools for Cyber-physical Systems. In Proceedings of the 6th International Workshop on Multi-Paradigm Modeling, Pages 49--54, ACM, New York, NY, USA, 2012.

[Brooks+2008] Christopher Brooks, Chihhong Cheng, Thomas Huining Feng, Edward A. Lee and Reinhard von Hanxleden. Model Engineering using Multimodeling. In 1st International Workshop on Model Co-Evolution and Consistency Management (MCCM '08), September 2008.

[Broy+2012] Manfred Broy, MaríaVictoria Cengarle and Eva Geisberger. Cyber-Physical Systems: Imminent Challenges. In Radu Calinescu and David Garlan editors, Large-Scale Complex IT Systems. Development, Operation and Management, Vol. 7539:1-28 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012.

[Becker+2006] Basil Becker, Dirk Beyer, Holger Giese, Florian Klein and Daniela Schilling. Symbolic Invariant Verification for Systems with Dynamic Structural Adaptation. In Proc. of the 28th International Conference on Software Engineering (ICSE), Shanghai, China, ACM Press, 2006.

[Becker&Giese2008] Basil Becker and Holger Giese. On Safe Service-Oriented Real-Time Coordination for Autonomous Vehicles. In In Proc. of 11th International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC), Pages 203--210, IEEE Computer Society Press, 5-7 May 2008.

[Becker&Giese2012] Basil Becker and Holger Giese. Cyber-Physical Systems with Dynamic Structure: Towards Modeling and Verification of Inductive Invariants. Technical report, 64, Hasso Plattner Institute at the University of Potsdam, Germany, 2012.

[Burmester+2008] Sven Burmester, Holger Giese, Eckehard Münch, Oliver Oberschelp, Florian Klein and Peter Scheideler. Tool Support for the Design of Self-Optimizing Mechatronic Multi-Agent Systems. In International Journal on Software Tools for Technology Transfer (STTT), Vol. 10(3):207-222, Springer Verlag, June 2008.

Bibliography (2/4)

[Dyck&Giese2017] Johannes Dyck and Holger Giese. K-Inductive Invariant Checking for Graph Transformation Systems. In Juan de Lara and Detlef Plump editors, Graph Transformation, Vol. 10373:142-158 of LNCS, Springer, Cham, 2017.

[Ghahremani+2018] Sona Ghahremani, Christian M. Adriano and Holger Giese. Training Prediction Models for Rule-Based Self-Adaptive Systems. In 2018 IEEE International Conference on Autonomic Computing (ICAC), Pages 187-192, 2018.

[Ghahremani+2017] Sona Ghahremani, Holger Giese and Thomas Vogel. Efficient Utility-Driven Self-Healing Employing Adaptation Rules for Large Dynamic Architectures. In 2017 IEEE International Conference on Autonomic Computing (ICAC), Pages 59-68, July 2017.

[Giese+2015] Holger Giese, Thomas Vogel and Sebastian Wätzoldt. Towards Smart Systems of Systems. In Mehdi Dastani and Marjan Sirjani editors, Proceedings of the 6th International Conference on Fundamentals of Software Engineering (FSEN '15), Vol. 9392:1--29 of Lecture Notes in Computer Science (LNCS), Springer, 2015.

[Giese+2019] Holger Giese, Maria Maximova, Lucas Sakizloglou and Sven Schneider. Metric Temporal Graph Logic over Typed Attributed Graphs. In Fundamental Approaches to Software Engineering - 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic April 6-11, 2019, Proceedings, Pages 282--298, 2019.

[Giese&Schäfer2013] Holger Giese and Wilhelm Schäfer. Model-Driven Development of Safe Self-Optimizing Mechatronic Systems with MechatronicUML. In Javier Camara, Rogério de Lemos, Carlo Ghezzi and AntÃ³nia Lopes editors, Assurances for Self-Adaptive Systems, Vol. 7740:152-186 of Lecture Notes in Computer Science (LNCS), Springer, January 2013.

[Ghezzi2012| Carlo Ghezzi. Evolution, Adaptation, and the Quest for Incrementality. In Radu Calinescu and David Garlan editors, Large-Scale Complex IT Systems. Development, Operation and Management, Vol. 7539:369-379 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012.

[Krause&Giese2012] Christian Krause and Holger Giese. Probabilistic Graph Transformation Systems. In Proceedings of Intern. Conf. on Graph Transformation (ICGT' 12), Vol. 7562:311-325 of Lecture Notes in Computer Science, Springer-Verlag, 2012.

Bibliography (3/4)

[Maier1998] Mark W. Maier. Architecting principles for systems-of-systems. In Systems Engineering, Vol. 1(4):267--284, John Wiley & Sons, Inc., 1998.

[Maximova+2018] Maria Maximova, Holger Giese and Christian Krause. Probabilistic Timed Graph Transformation Systems. In Journal of Logical and Algebraic Methods in Programming, Vol. 101:110 - 131, 2018.

[Maximova+2021] Maria Maximova, Sven Schneider and Holger Giese. Compositional Analysis of Probabilistic Timed Graph Transformation Systems. In Proc. Of Fundamental Approaches to Software Engineering (FASE 2021), Luxembourg, March 27 - April 1, 2021, , Vol. 12649:196--217 of Lecture Notes in Computer Science, Springer, 2021.

[Northrop+2006] Northrop, Linda, et al. Ultra-Large-Scale Systems: The Software Challenge of the Future. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.

[Pereira+2013] Eloi Pereira, Christoph M. Kirsch, Raja Sengupta and Jo~ao Borges de Sousa. Bigactors - A Model for Structure-aware Computation. In ACM/IEEE 4th International Conference on Cyber-Physical Systems, Pages 199--208, ACM/IEEE, Philadelphia, PA, USA, 2013.

[Sakizloglou+2020| Lucas Sakizloglou, Sona Ghahremani, Thomas Brand, Matthias Barkowsky and Holger Giese. Towards Highly Scalable Runtime Models with History. In 15th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2020, Seoul South Korea, October, 2020, IEEE Computer Society, 2020.

[Sakizloglou+2022| Lucas Sakizloglou, Sona Ghahremani, Matthias Barkowsky, Holger Giese. Incremental execution of temporal graph queries over runtime models with history and its applications. In Software and Systems Modeling, Vol. 21, 2022.

[Schneider+2020] Sven Schneider, Johannes Dyck and Holger Giese. Formal Verification of Invariants for Attributed Graph Transformation Systems Based on Nested Attributed Graph Conditions. In Fabio Gadducci and Timo Kehrer editors, Graph Transformation - 13th International Conference, ICGT 2020 Held as Part of STAF 2020, Bergen, Norway, June 25-26, 2020, Proceedings, Vol. 12150:257--275 of Lecture Notes in Computer Science, Springer, 2020.

Bibliography (4/4)

[Sztipanovits2011] Janos Sztipanovits with Ted Bapty, Gabor Karsai and Sandeep Neema. MODEL-INTEGRATION AND CYBER PHYSICAL SYSTEMS: A SEMANTICS PERSPECTIVE. FM 2011, Limerick, Ireland. 22 June 2011

[Sztipanovits+2012] Janos Sztipanovits, Xenofon Koutsoukos, Gabor Karsai, Nicholas Kottenstette, Panos Antsaklis, Vineet Gupta, B. Goodwine, J. Baras and Shige Wang. Toward a Science of Cyber-Physical System Integration. In Proceedings of the IEEE, Vol. 100(1):29-44, January 2012.

[Valerdi+2008] Ricardo Valerdi, Elliot Axelband, Thomas Baehren, Barry Boehm, Dave Dorenbos, Scott Jackson, Azad Madni, Gerald Nadler, Paul Robitaille and Stan Settles. A research agenda for systems of systems architecting. In International Journal of System of Systems Engineering, Vol. 1(1-2):171--188, 2008.

[Vogel+2009] Thomas Vogel, Stefan Neumann, Stephan Hildebrandt, Holger Giese and Basil Becker: Model-Driven Architectural Monitoring and Adaptation for Autonomic Systems. In: Proc. of the 6th International Conference on Autonomic Computing and Communications (ICAC'09), Barcelona, Spain, ACM (15-19 June 2009)

[Vogel+2010] Thomas Vogel and Stefan Neumann and Stephan Hildebrandt and Holger Giese and Basil Becker. Incremental Model Synchronization for Efficient Run-Time Monitoring. In Sudipto Ghosh, ed., Models in Software Engineering, Workshops and Symposia at MODELS 2009, Denver, CO, USA, October 4-9, 2009, Reports and Revised Selected Papers, vol. 6002 of Lecture Notes in Computer Science (LNCS), pages 124-139. Springer-Verlag, 4 2010.

[Vogel&Giese2012] Thomas Vogel and Holger Giese. A Language for Feedback Loops in Self-Adaptive Systems: Executable Runtime Megamodels. In Proceedings of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2012), pages 129-138, 6 2012. IEEE Computer Society.

[Zöllner+2021] Christian Zöllner, Matthias Barkowsky, Maria Maximova, Holger Giese. On the Complexity of Simulating Probabilistic Timed Graph Transformation Systems. In Graph Transformation - 14th International Conference, ICGT 2021 Held as Part of STAF 2021, Virtual Event, June 24-25, 2021, Proceedings (Fabio Gadducci, Timo Kehrer, eds.), Springer, volume 12741, 2021.