Actuation throughout DevOps

Jean-Yves Tigli, UCA, CNRS
S. Lavirotte, UCA, CNRS

WP2 – T2.3
WP3 – T3.2

www.enact-project.eu

Actuation challenges

- Most of the IoT platforms and applications consider actuation as :
  - Inexistent (sensor networks),
  - "Fire & Forget" (actuation is considered unproblematic *),
  - Ad-hoc actuation management**.

Call IoT-03-2017 : R&I on IoT integration and platforms

"IoT platforms integrating evolving sensing, actuating, "

"Platforms should provide connectivity and intelligence, actuation and control features"

Trustworthlyness** is **actuation management and control

(from a semantic point of view of action and conflictual effects control)

ENACT DevOps

** Purely programmatic approach*

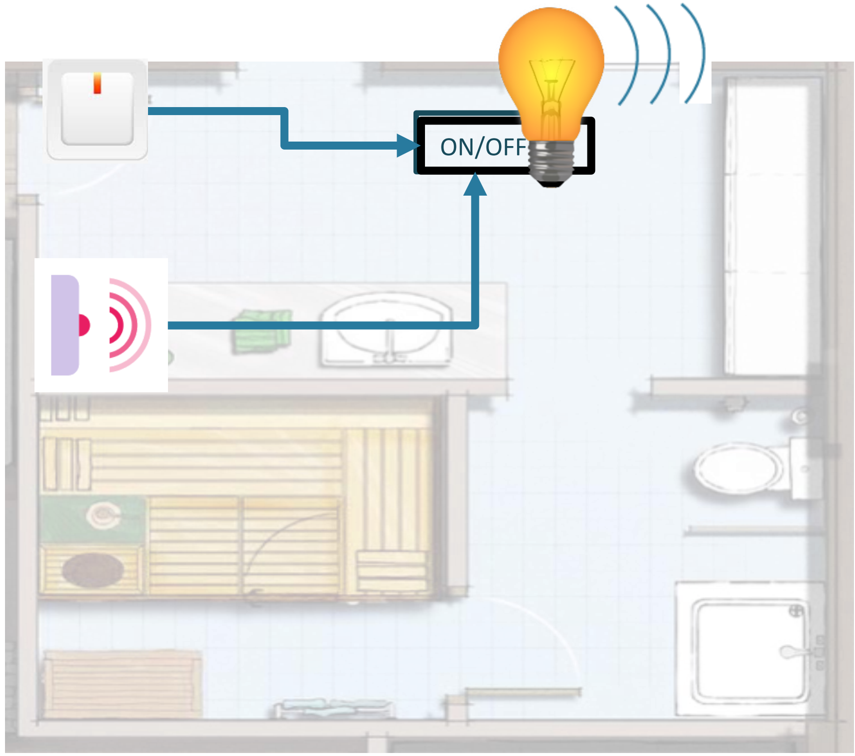
*** Industrial automatic control, embedded systems, etc…*

Actuation Conflict Handling (T2.3)

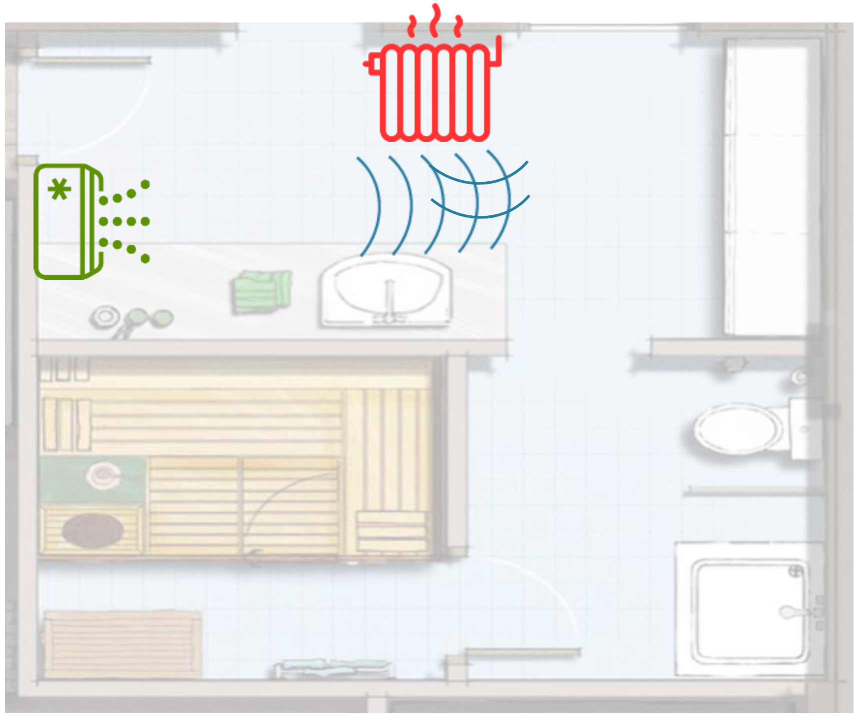
Key idea : « Actuation conflict is not only action conflict but also physical effect conflict »

Necessary physical effect model

- To care of Explicit but also Implicit Effects (a.k.a. direct and indirect impacts [Yagita et al.])
- Only few works consider implicit actuation effects (see related works D2.1 and D3.1)



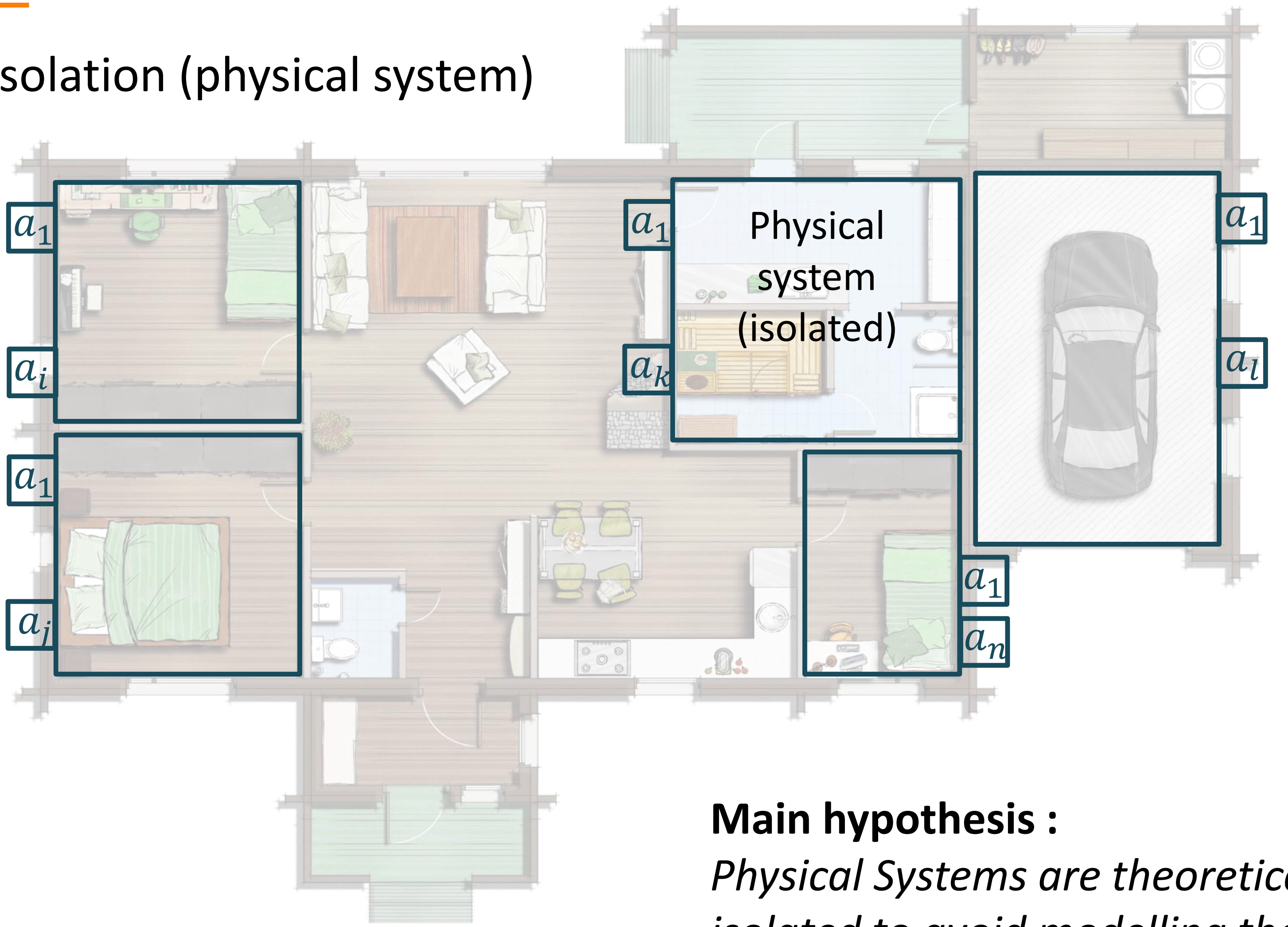
Explicit conflict



Implicit conflict

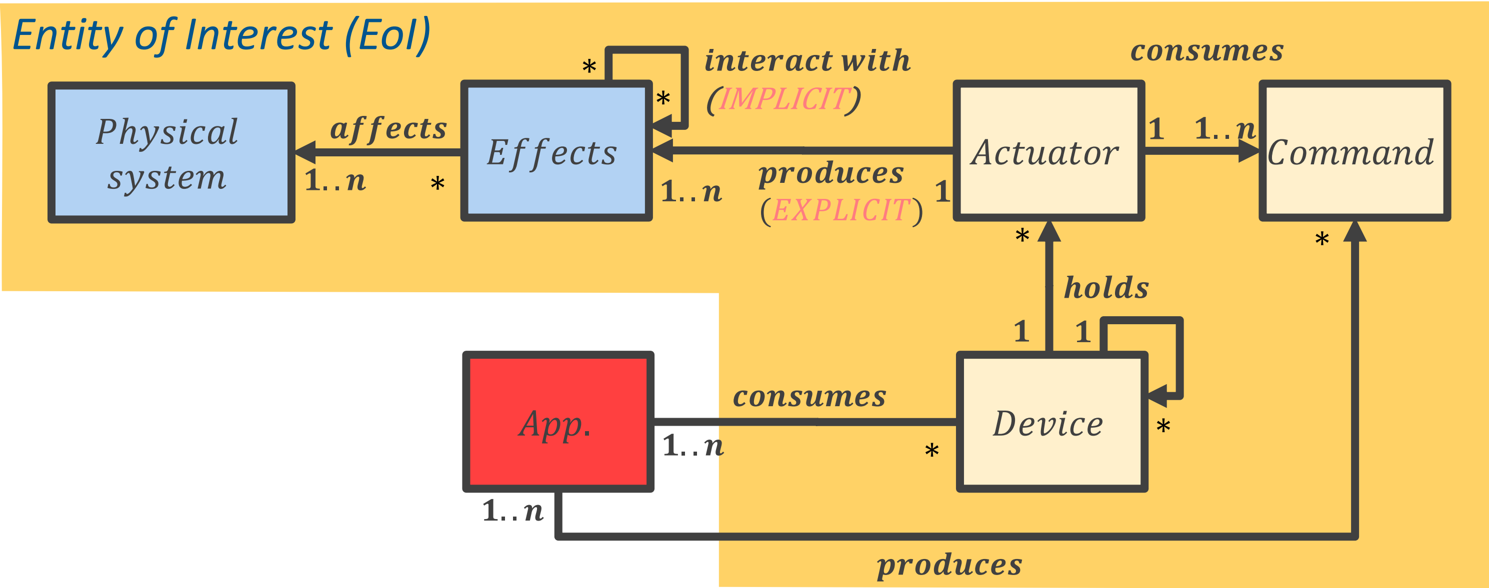
Physical system model is key to model effects

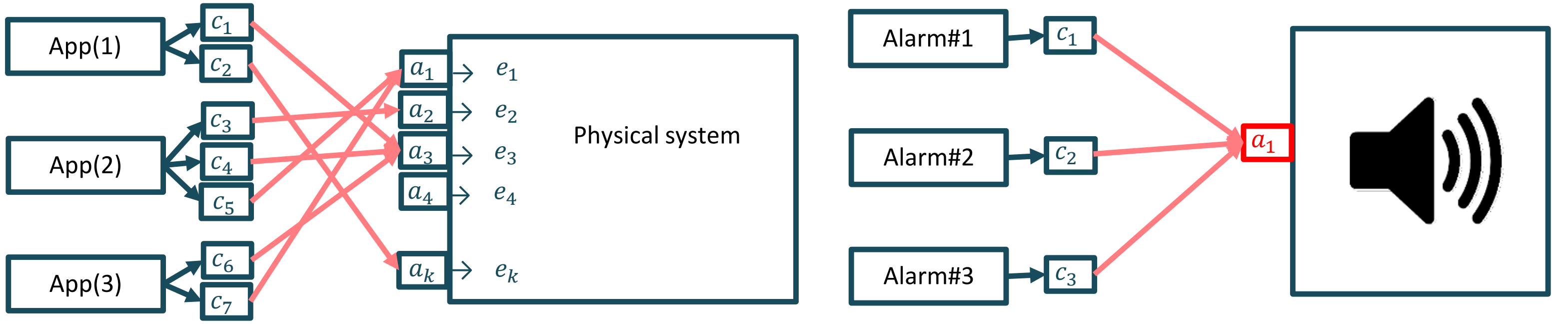
- Actuation impacts isolation (physical system)



Main hypothesis :
Physical Systems are theoretically isolated to avoid modelling the world

Action-Effect model





Entity of interest

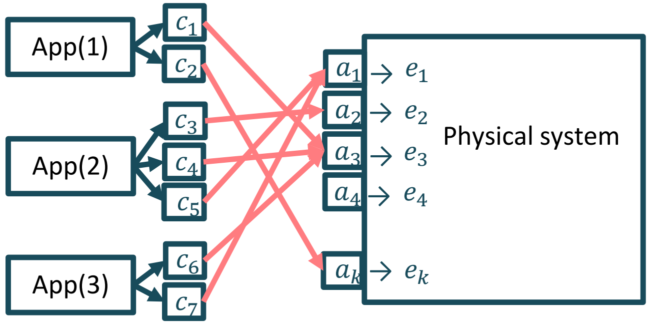
[Haller S.]

[Sarray I. et al.]

Also close to other models [Zhao_b et al.]

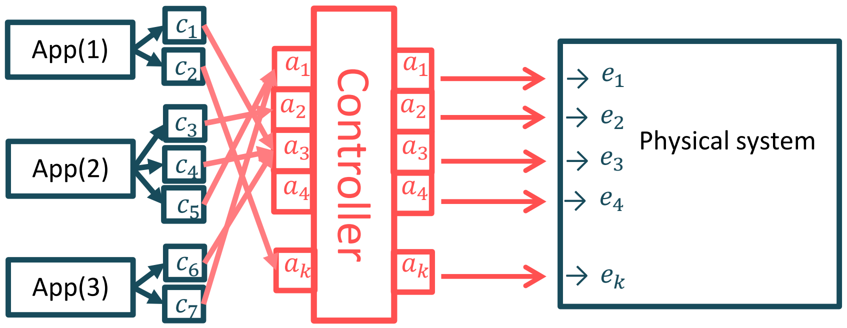
ENACT DevOps

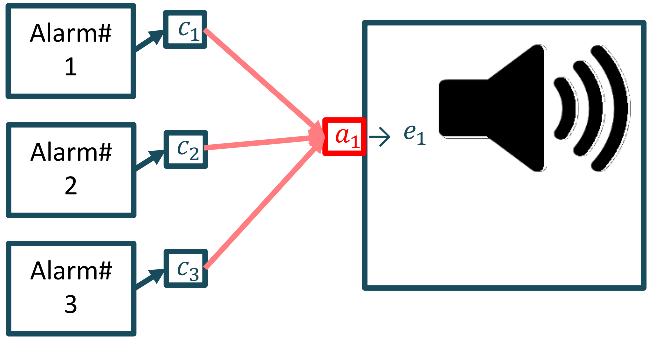
T2.3 challenge : Actuation Controller Model and Synthesis

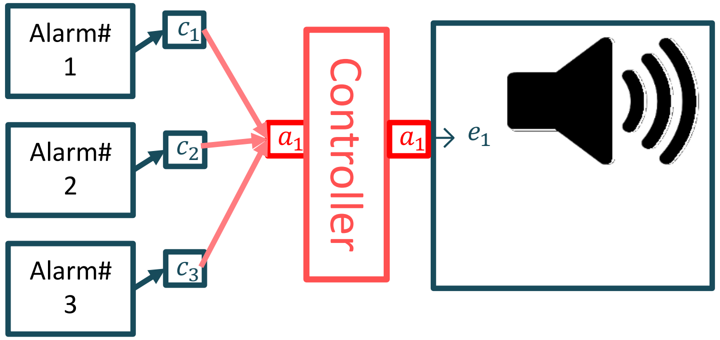


Adding constraints model to Physical System and Orchestration Models

Designing controller with a model checker





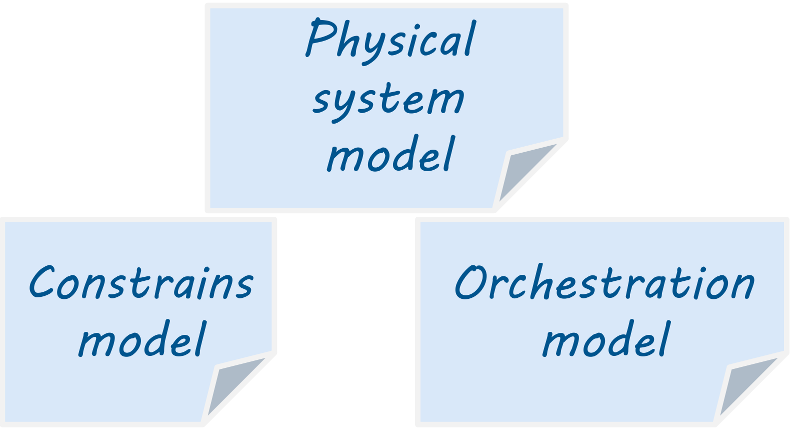


ENACT DevOps

www.enact-project.eu

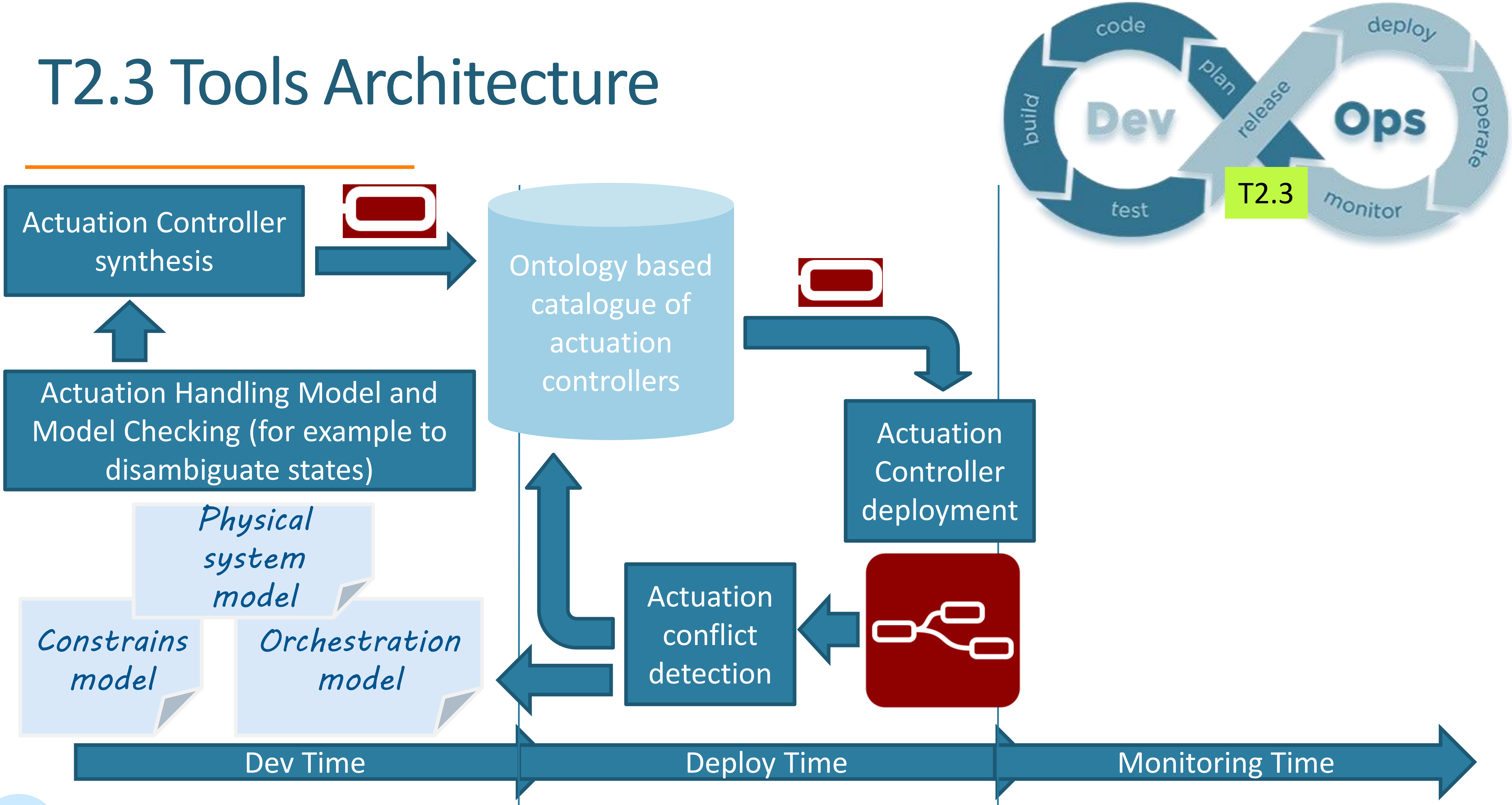
CNRS contribution in T2.3

- Models Definition :
  - Definition of models for actuation & physical systems
  - Orchestration model and actuation conflict checking
  - Constrains model on concurrent accesses



- Actuation Controllers Design & Storage :
  - Assisted design of actuation handling with constrains specification (model checking)
  - Storage of actuation controller in an ontology-based catalogue
  - Synthesis of actuation controller to deploy dynamically

T2.3 Tools Architecture



Actuation and behavioral drift (T3.2)

Key Idea : "what is effectively done is not what is expected"

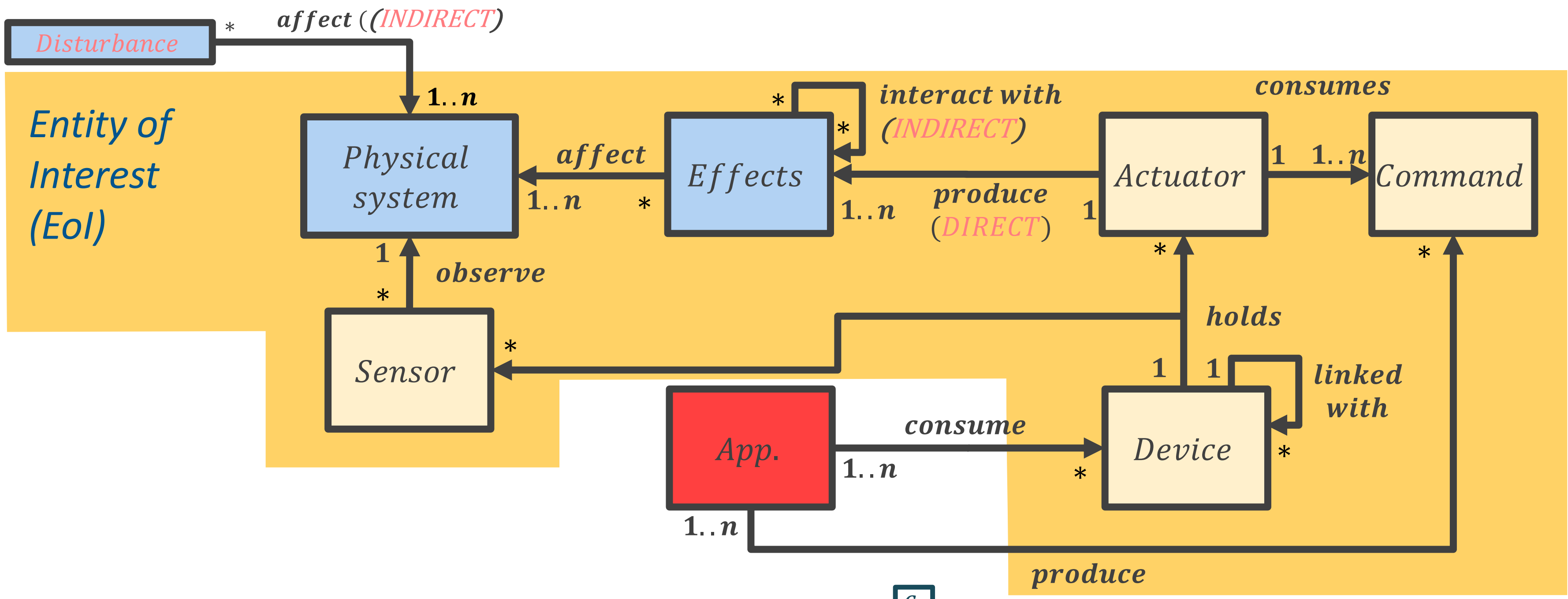
However,
"what is effectively done is not what is expected"

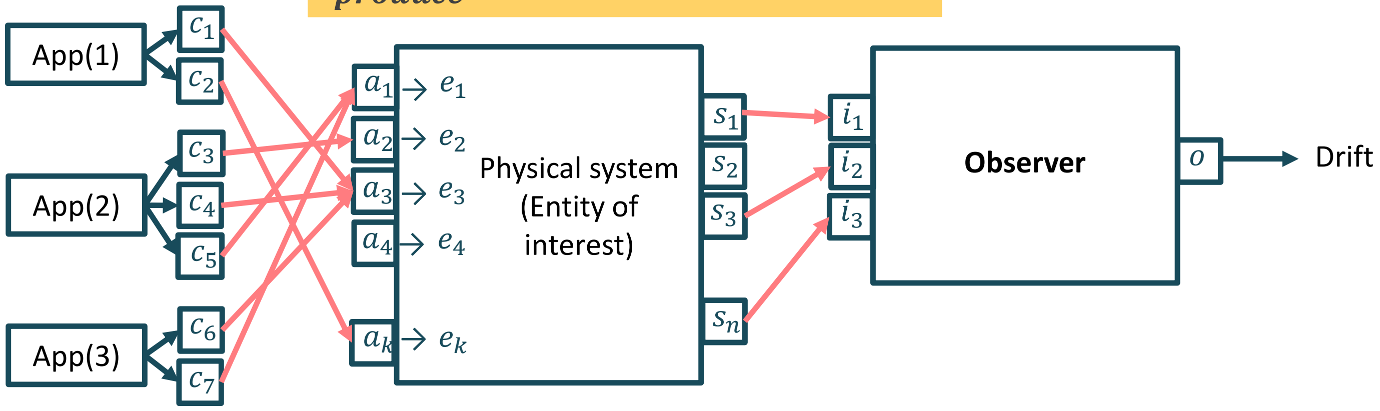
- From T2.3 to T3.2
- Physical systems are subject to uncertainties,
- Indirect effects may cause unexpected behaviours.



ENACT DevOps

Action-effect-observation model

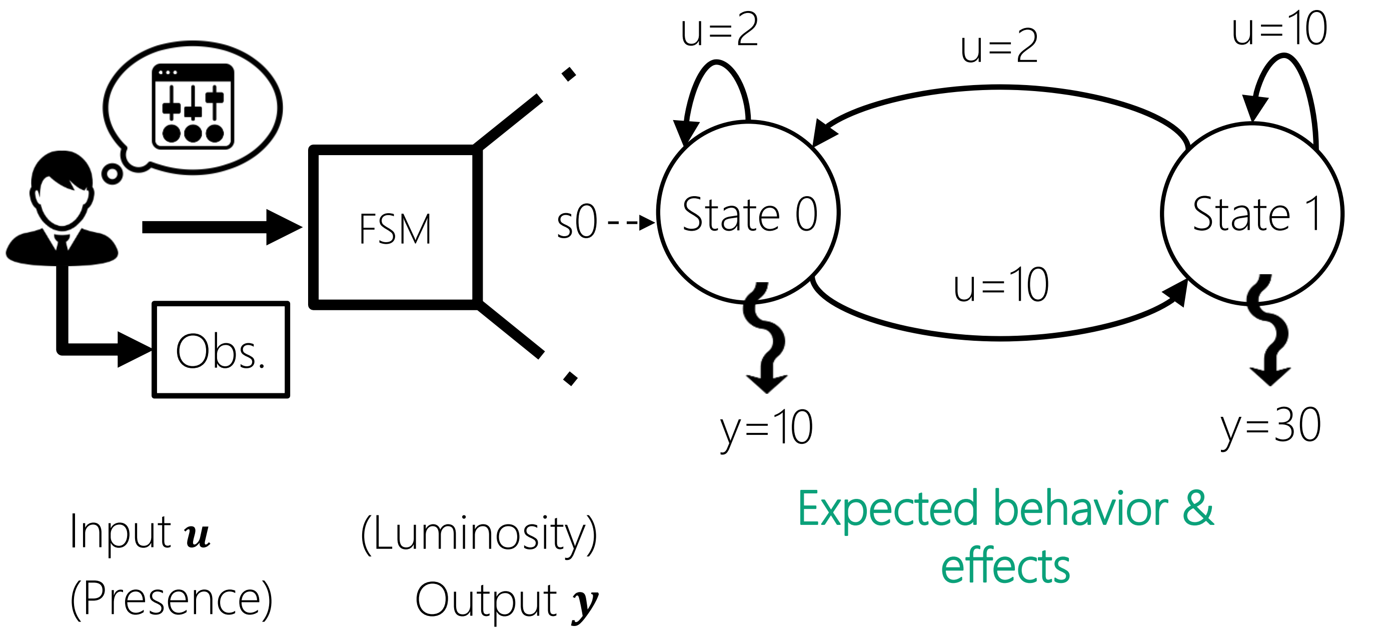




Intuitive approach : Deterministic observation Model

	y=10 Lux (State 0)	y=30 Lux (State 1)
Presence (u=2)	Conform	Non–Conform
Presence (u=10)	Non-Conform	Conform

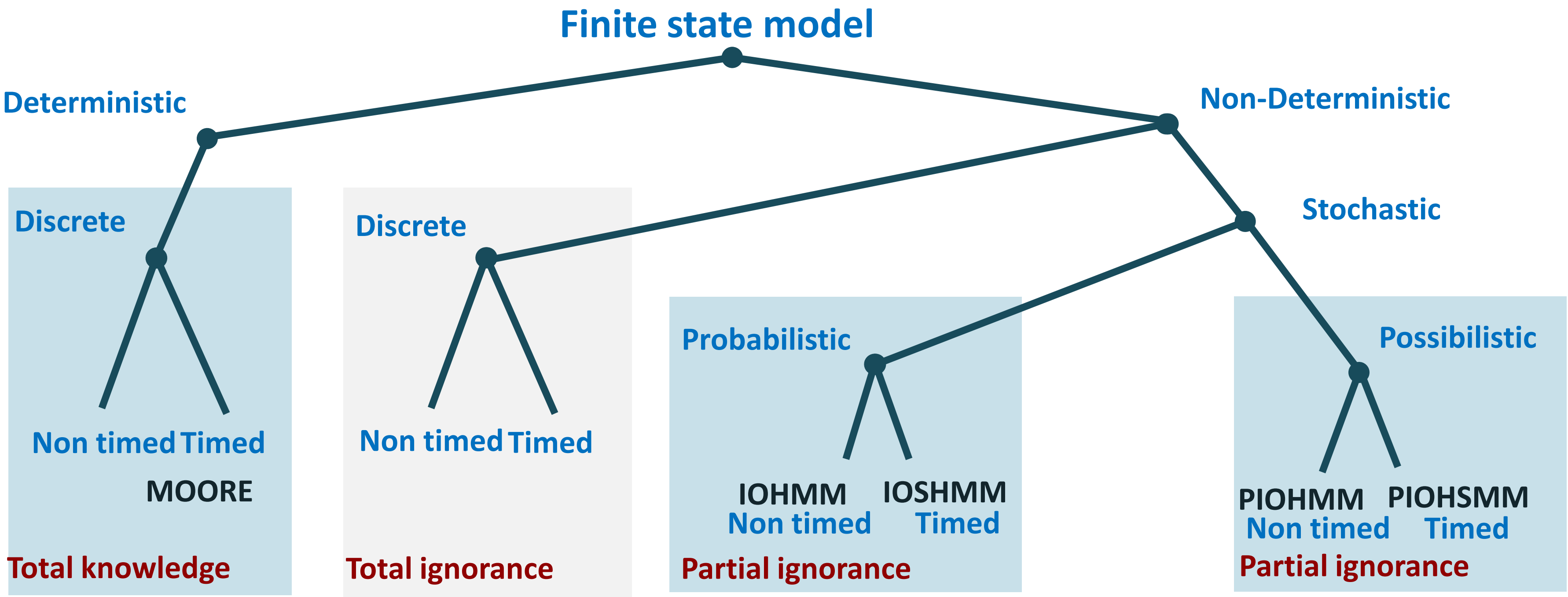
However Illusory Conformity !



Main approach : From deterministic model to stochastic model to handling uncertainties

- From a **deterministic model of the expected behaviour**, free from uncertainties,
- To **stochastic model handling uncertainties/vagueness**

T3.2 : Exploration of new associated deterministic / stochastic models

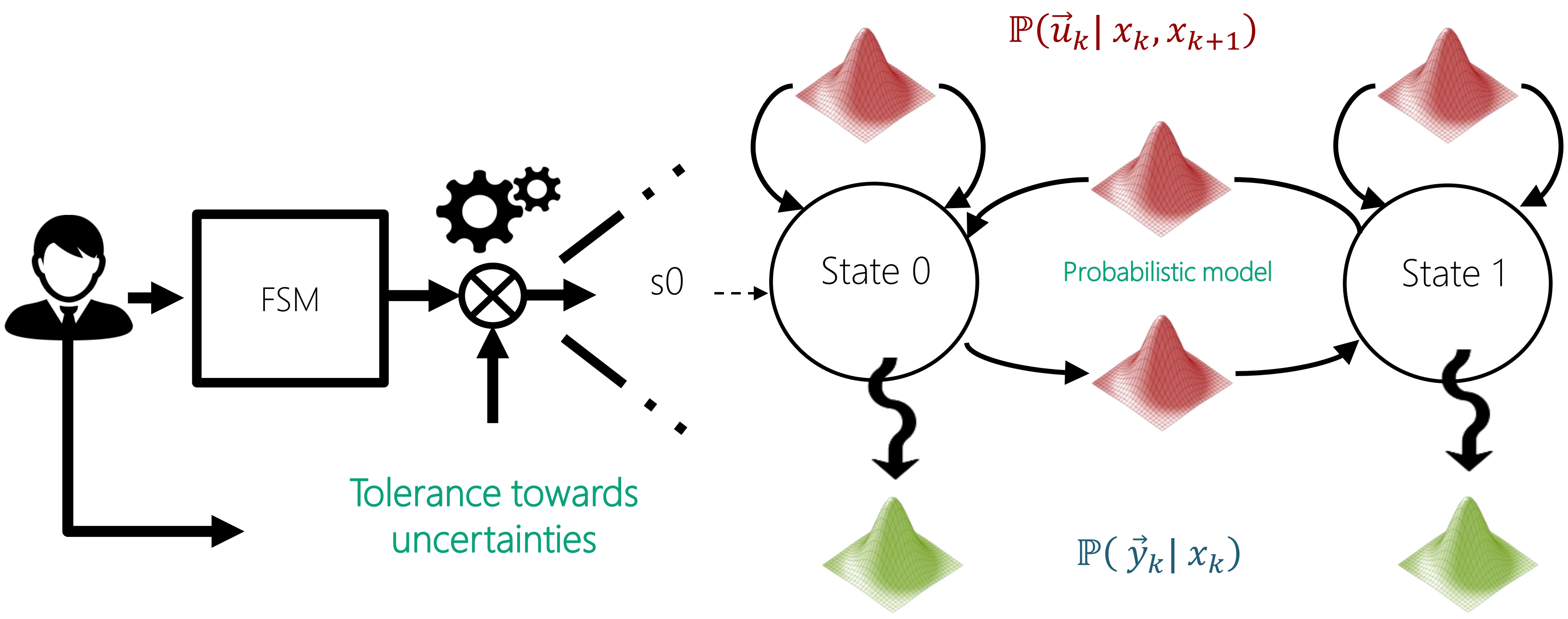


ENACT DevOps

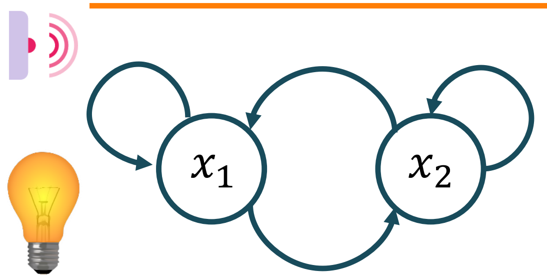
www.enact-project.eu

Stochastic Observer Design and Synthesis

- Example Probabilistic Observer Model



Stochastic Observer Design and Synthesis



Expected deterministic observer model light/presence

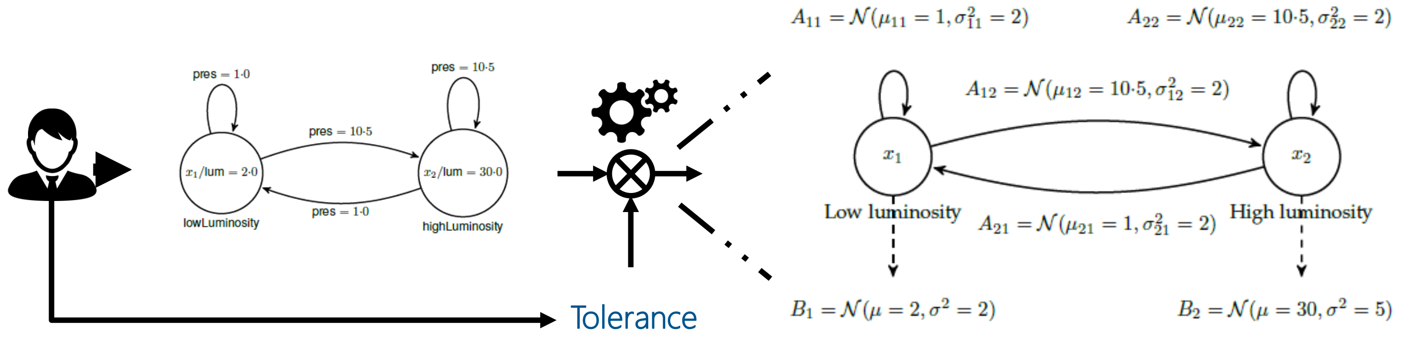


Fig.a : Input/Output observations

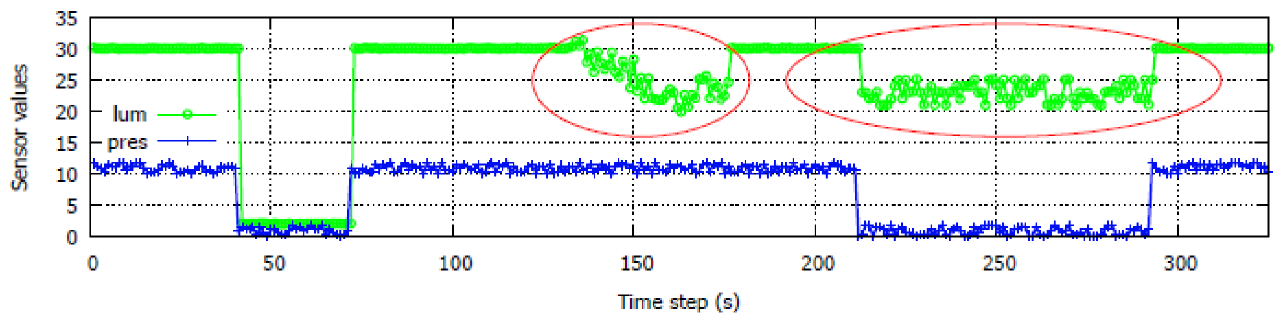
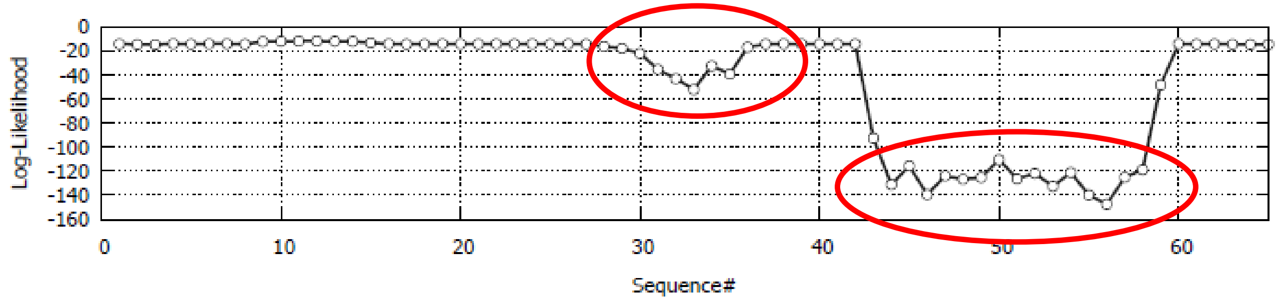


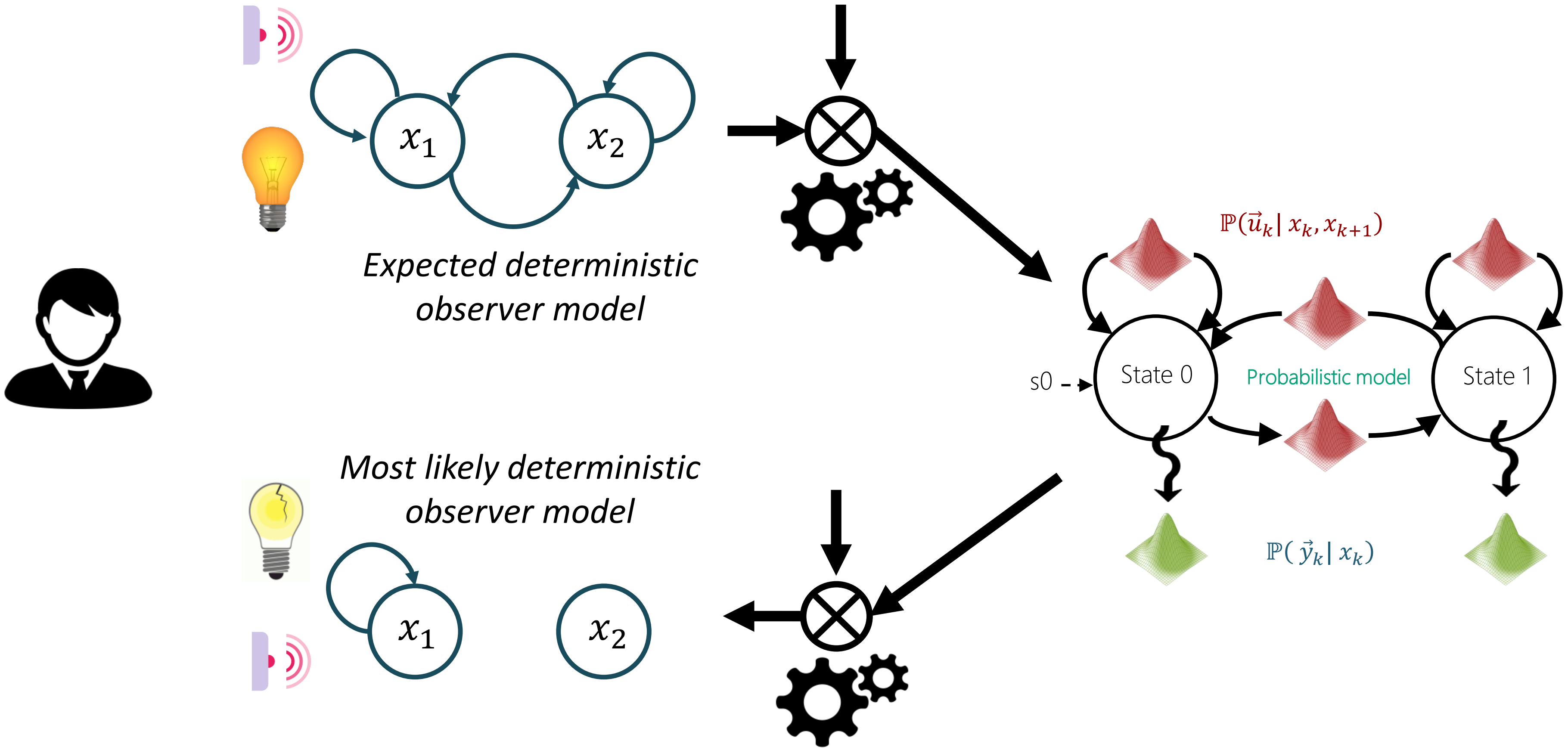
Fig.b : Log-Likelihood
with Input/Output sequences length (K) = 5



Smart home example

ENACT DevOps

Most likely deterministic observer model as feedback

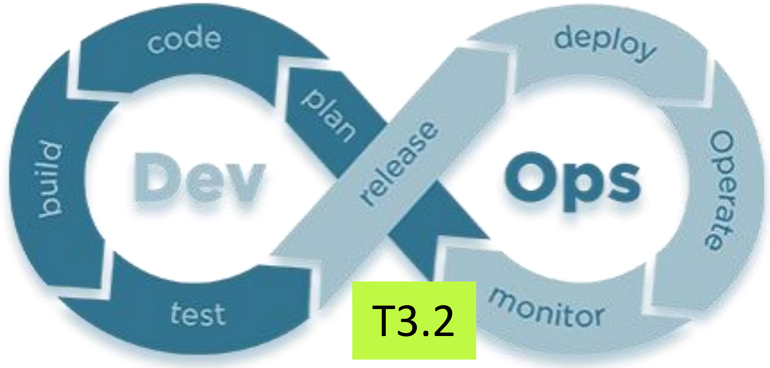


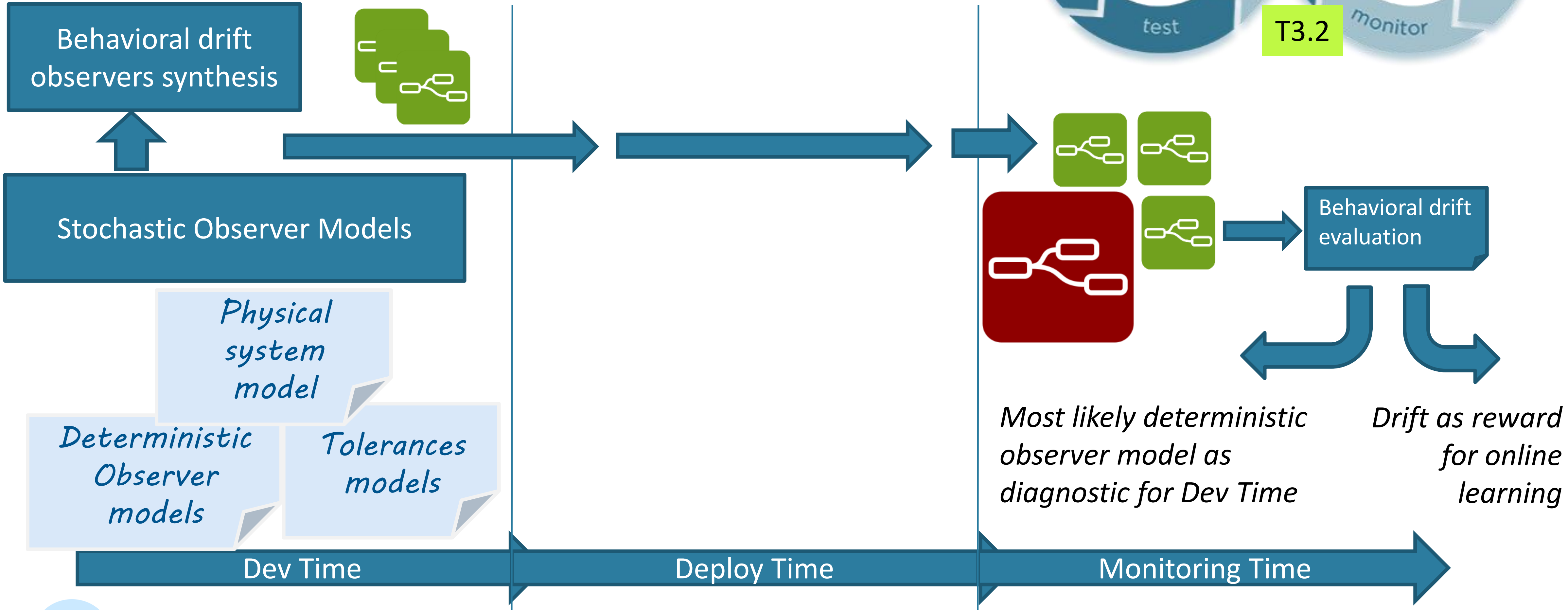
CNRS contribution in T3.2

- Context observers providing Stochastic Model inputs (to discuss with UDE)
- Exploration of new associated deterministic / stochastic models (CNRS)

- Test Behavioral Drifts measures as reward for reinforcement learning (UDE)
- Feedback for « Dev » :
  - Behavioral drift
  - The most likely observers deterministic model

T3.2 Tools Architecture





Experiments with use case providers

Experiments in Smart Building Domain with Tecnalia

- Step 1 : Scenarios Description
- Step 2 : Requirements and Experimental Infrastructure
- Step 3 : Applications Design to illustrate actuation conflict challenge :
  - At Dev Time : Actuation conflict handling
  - At Ops Time : Behavioral drift evaluation
- Step 4 : Continuous delivery improvment
  - Experiments and experimental protocols
  - Feedback analysis (what are benchmarks ?)

	Actions	Observations
Confort	Heater	Temperature Sensors
Luminosity	Controllable windows – Controllable light	Light sensors (inside / outside)

- Others use cases in ITS Domain (Rail with Indra) ?

Questions ?

Appendices

Appendice 1 : Bibliography

Bharathan et al.	Bharathan Balaji, Bradford Campbell, Amit Levy, Xiaozhou Li, Addison Mayberry, Nirupam Roy, Vasuki Narasimha Swamy, Longqi Yang, Victor Bahl, Ranveer Chandra, Ratul Mahajan: Modeling Actuation Constraints for IoT Applications. CoRR abs/1701.01894 (2017)
Zhao_a et al.	Mengxuan Zhao, Gilles Privat, Éric Rutten, Hassane Alla: Discrete Control for Smart Environments Through a Generic Finite-State-Models-Based Infrastructure. AmI 2014: 174-190
Zhao_b et al.	Mengxuan Zhao, Gilles Privat, Éric Rutten, Hassane Alla: Discrete Control for the Internet of Things and Smart Environments. Feedback Computing 2013
Munir et al.	Sirajum Munir, John A. Stankovic: DepSys: Dependency aware integration of cyber-physical systems for smart homes. ICCPS 2014: 127-138
Otani et al.	Masayuki Otani, Toru Ishida, Yohei Murakami, Takao Nakaguchi: Event management for simultaneous actions in the Internet of Things. WF-IoT 2016: 64-69
Sarray et al.	Ines Sarray, Annie Ressouche, Daniel Gaffé, Jean-Yves Tigli, Stephane Lavirotte: Safe Composition in Middleware for the Internet of Things. M4IoT@Middleware 2015: 7-12
Ressouche et al.	Annie Ressouche, JY Tigli, and Oscar Carrillo: Towards Validated Composition in Component-Based Adaptive Middleware. In Software Composition, 165–180 (2011)
Takeuchi et al.	Susumu Takeuchi, Michiharu Takemoto, Masato Matsuo: SPIRE: Scalable and Unified Platform for Real World IoT Services with Feature Interaction. COMPSAC Workshops 2016: 348-353
Umakoshi et al.	Kenji Umakoshi, Takashi Kambayashi, Manabu Yoshida, Michiharu Takemoto, Masato Matsuo: S3: Smart Shadow System for Real World Service and Its Evaluation with Users. SAINT 2011: 394-401
Haller, S.	Haller, S., The things in the internet of things, Internet of Things Conference 2010, Tokyo, Japan.
Yagita et al.	Miki Yagita, Fuyuki Ishikawa, Shinichi Honiden: An Application Conflict Detection and Resolution System for Smart Homes. SEsCPS@ICSE 2015: 33-39

Appendice 1 : First related works analysis

Publications	Ontology based	Behavioural model	Dependency graph	Controler model	Constraints model	Direct effects	Indirect effects	Conflicts mgmt
Modeling Actuation Constraints for IoT Applications (3 pages position paper)	Yes		Yes					
Discrete Control for Smart Environments through a Generic Finite State Models Based Infrastructure	Yes	FSM (Devices)	Yes	FSMs Composition	Boolean expressions	Yes	NO	Garenteed by **design**
DepSys: Dependency Aware Integration of Cyber-Physical Systems for Smart Homes	No	(APPs level) requirements manifest	Yes		Actuators/sensors Requirements manifest	yes	NO	**Deployment**, **Run-time**
Event Management for Simultaneous Actions in the Internet of Things	No		Yes	// rules execution mgmt.	Complex Event Processing (CEP)	Yes	NO	**Run-time**
Safe Composition in Middleware for the Internet of Things	No	Mealy machines (devices level)	no	FSMs Composition	Description constraint language (DCL)	yes	NO	**Run-time**
SPIRE: Scalable and Unified Platform for Real World IoT Services with Feature Interaction	No	ECA rules (Services level)	NO	Agents based	ECA rules	Yes	NO	**Run-time** Features interactions (no implem.)
An Application Conflict Detection and Resolution System for Smart Homes	No	System Metadata parser (apps,actuators, sensors)	no	Checker/resolver (Kripke structure to detect conflicts by model-checking the assertion "no two apps use actuators to create different effects)		Yes	NO	**Deployment** User to approve

DevOps