Ambient Intelligence

<< More than the sum of its devices, the Internet of Things [Ambient Intelligence] links technologies together to create new services and opportunities. >>

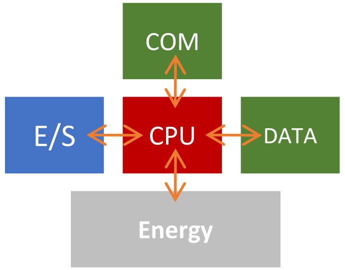


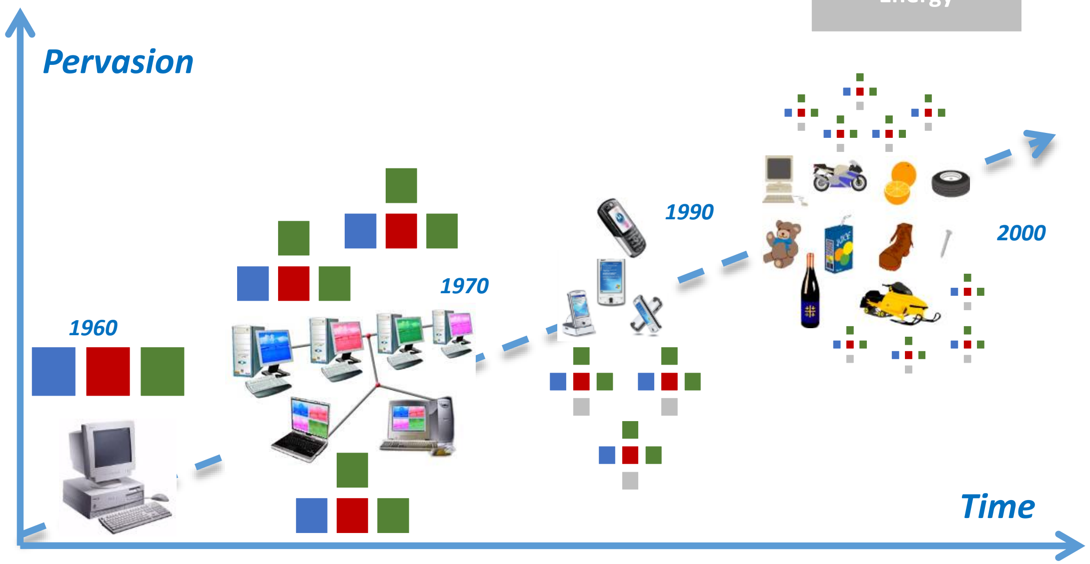

Ubiquitous Computing (1991)

« Silicon-based information technology, is far from having become part of the environment »

- Everytime, Everywhere, but in Everything
- Ubiquitous Computing is a Post distributed Distributed Computing
- After networks of distributed computers, mobiles computers, it's time for distributed things and smart objects

From Von Neumann Computer Model to Smart Objects ....





Our approach : OPPORTUNISTIC Sofware

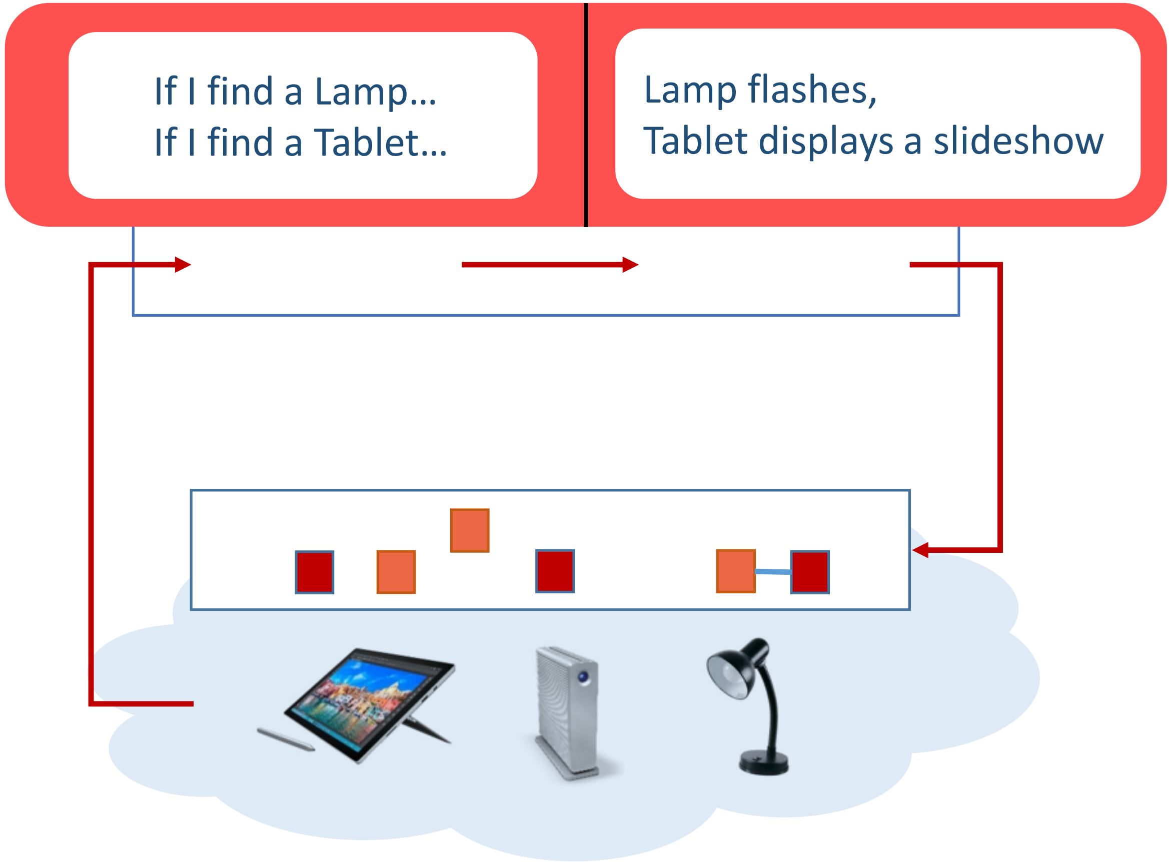
Example : Assisted Living for elderly people

- Assisted Living, new services for elderly people
- Here is a way to provide solicitation service for an apathic person
- Best effort means Opportunistic Software

Opportunistic Software : Principles and Challenges (1/2)

- Application Models as a pair of rules to:
  - Dynamically find among all the available web service on devices (WSD), those relevant for a specific application,
  - Compose these WSD to provide such a specific application

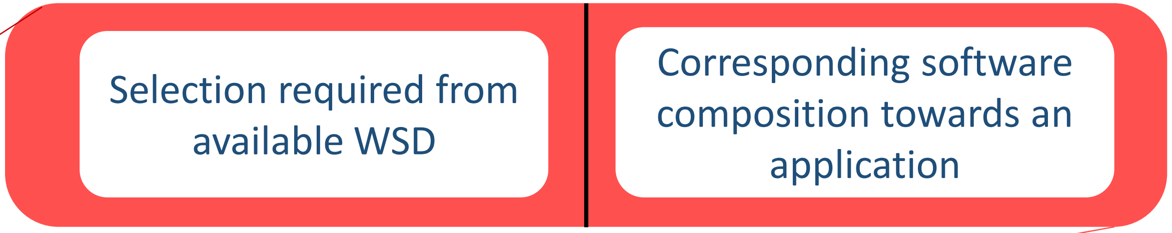
« Best effort » to deploy an application:
- *flashing a lamp*
- *displaying a diaporama on a tablet*

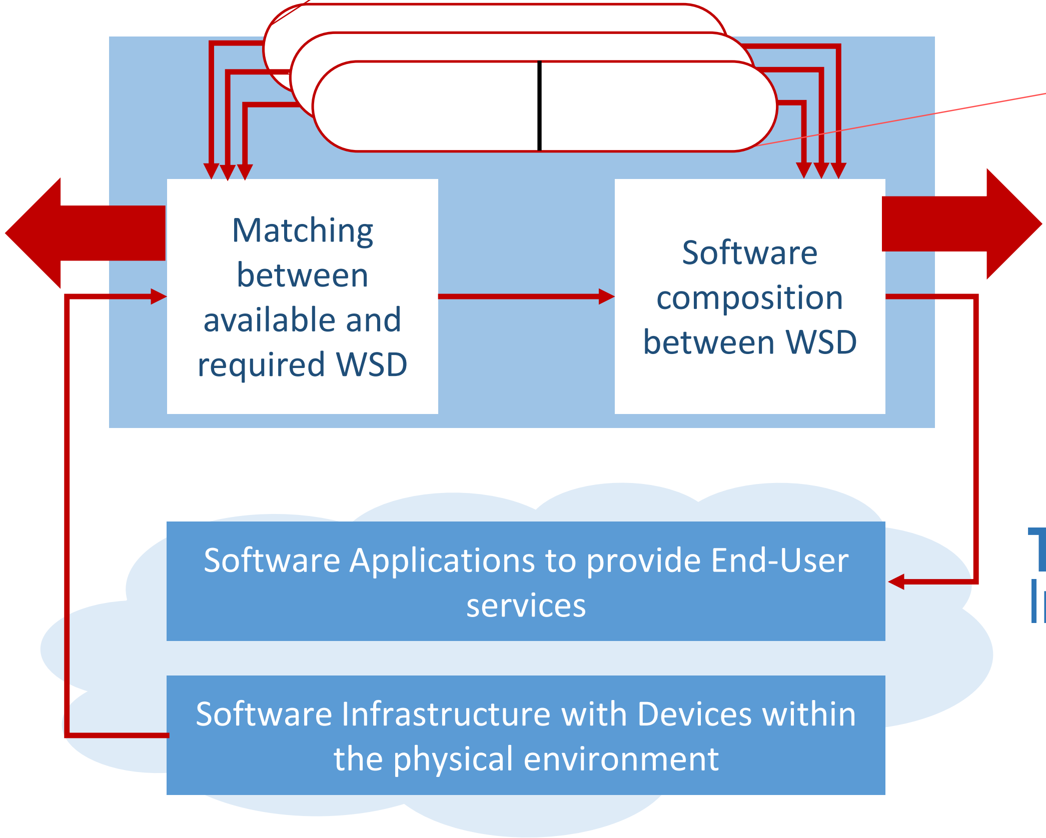


Opportunistic Software : Principles and Challenges

Principles : Opportunistic Composition

- Models of Applications as a pair of rules





1 **Semantic Matching**
Heterogeneity *and*
Interoperability

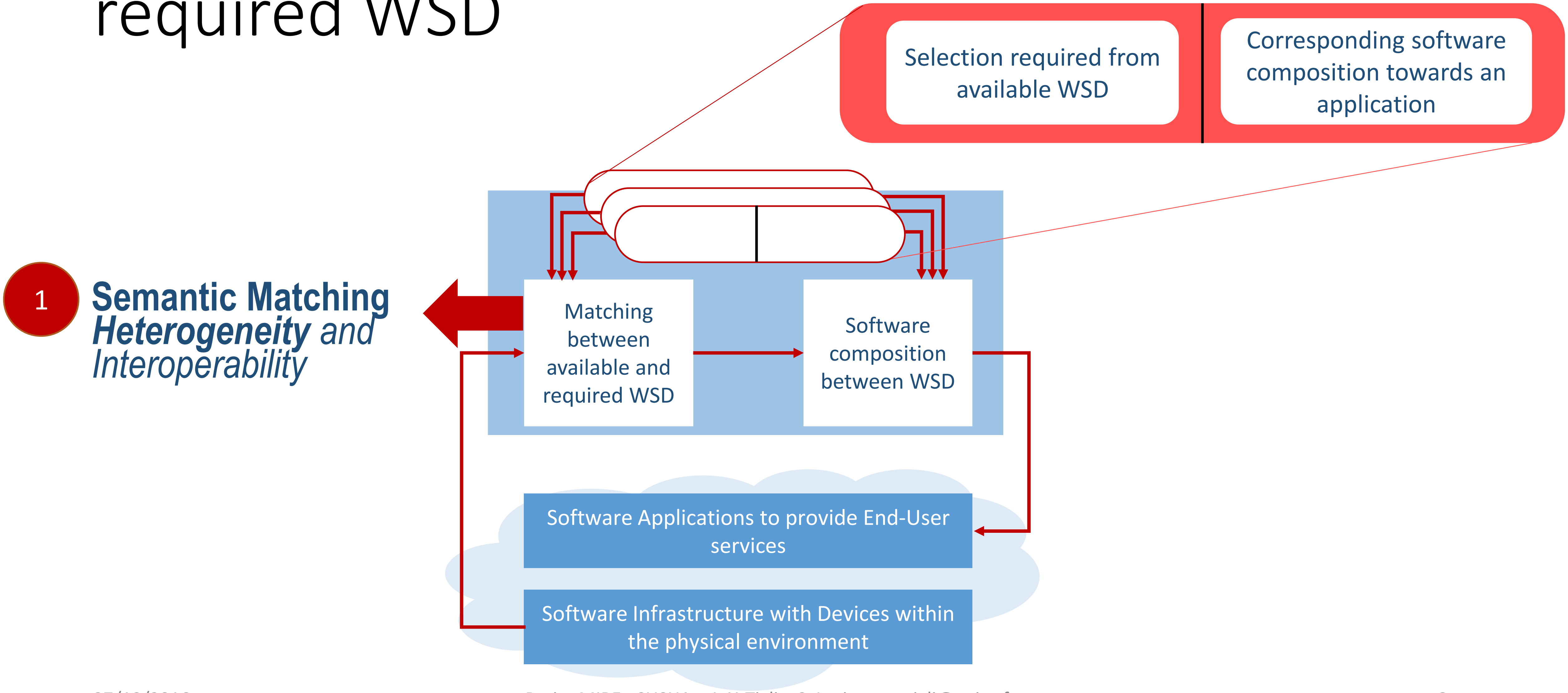
2 **Software Composition of multiple applications sharing physical devices**

Technological heterogeneity and Interoperability :

WoT and **WSD** : **Web Service** for Things (WoT) and **Device** (WSD)

First Challenge : Semantic Matching between available and required services on devices

Semantic Matching between available and required WSD

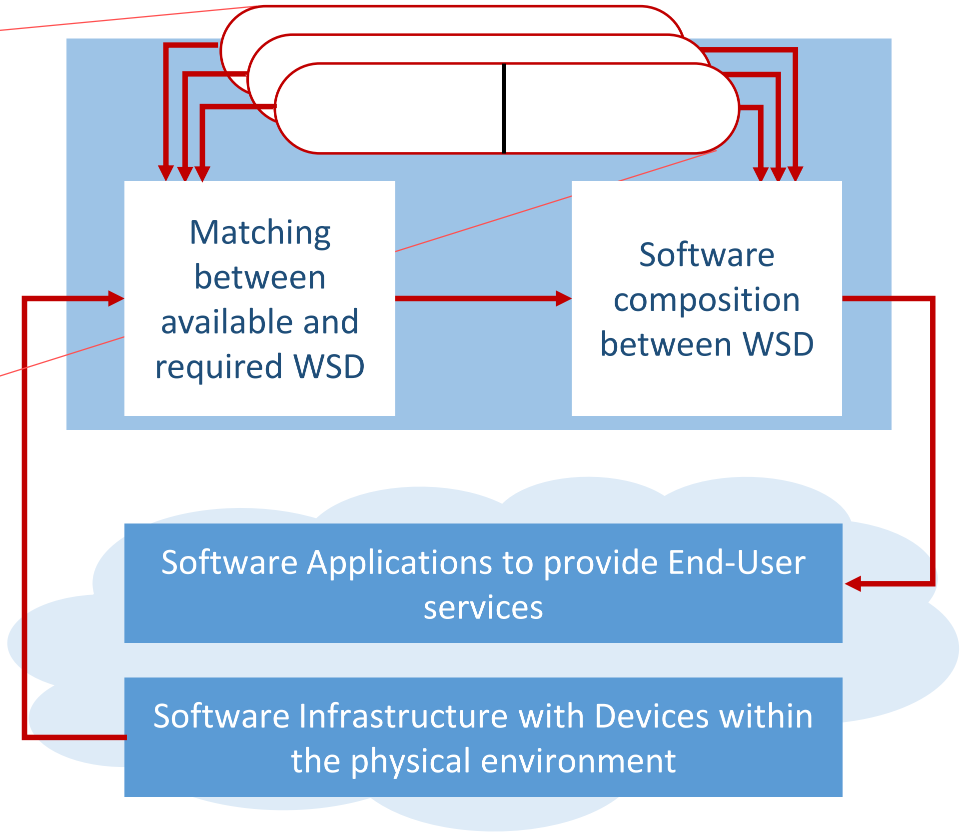


Our first approach

Syntactic alignment between available WSD

LAMP : Lamp*
DISPLAY : Tablet*
NAS : Nas*

```
LAMP.switch()
DISPLAY.slideshow(
NAS.File("C:/test.sld")
)
```



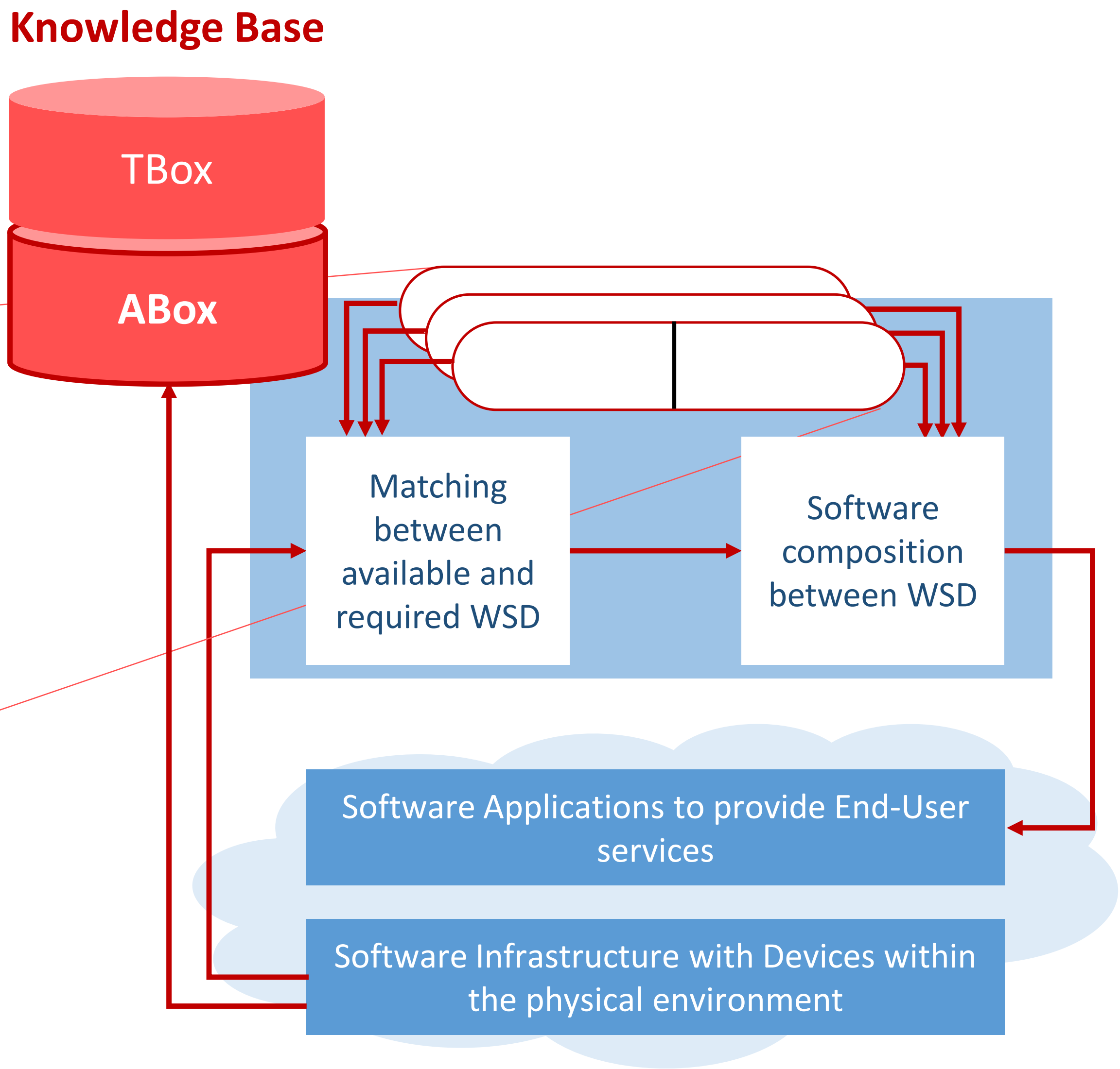
Matching is based on regular expressions on naming

Experimental Platform where we go (1/2)

WSD formally described from a common ontology (TBox)

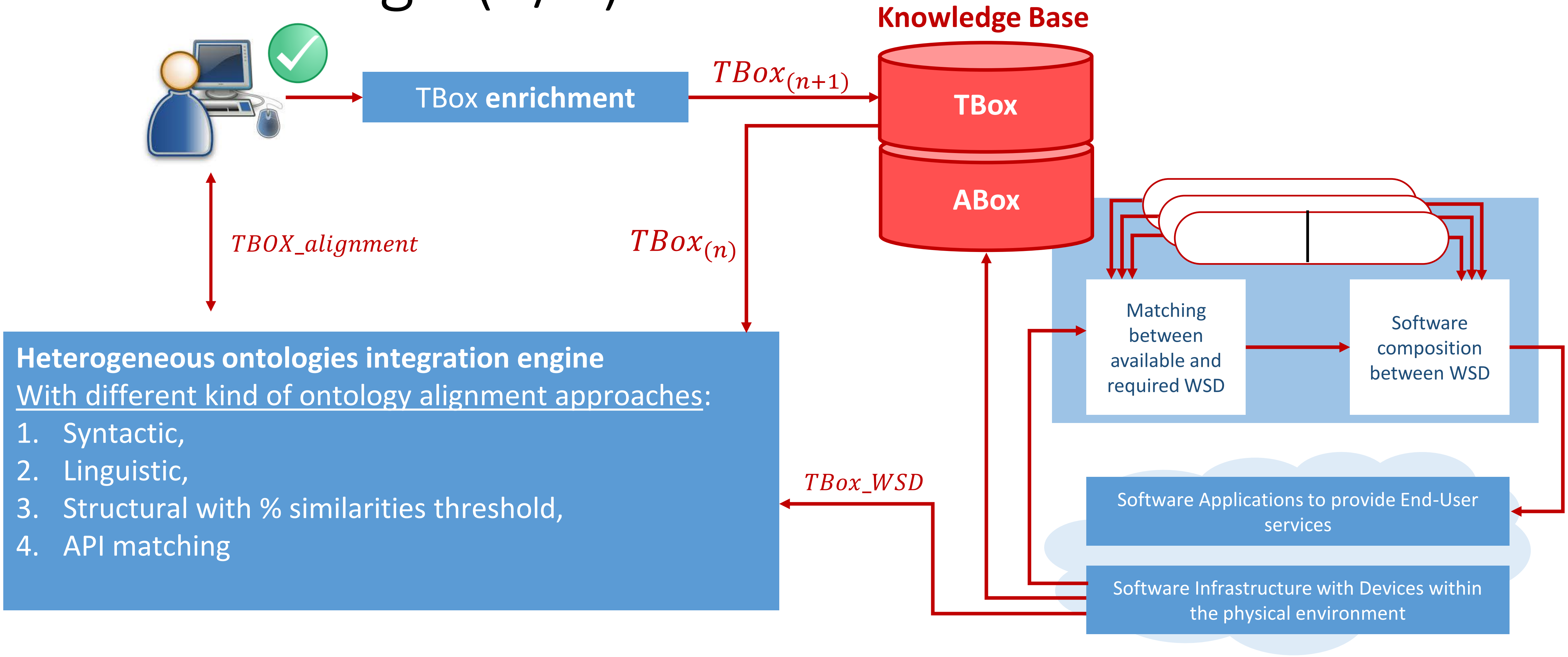
SPARQL queries !!!!

```
LAMP.switch()
DISPLAY.slideshow(
NAS.File("C:/test.sld")
)
```



Matching is based on SPARQL queries

Experimental Platform where we go (2/2)



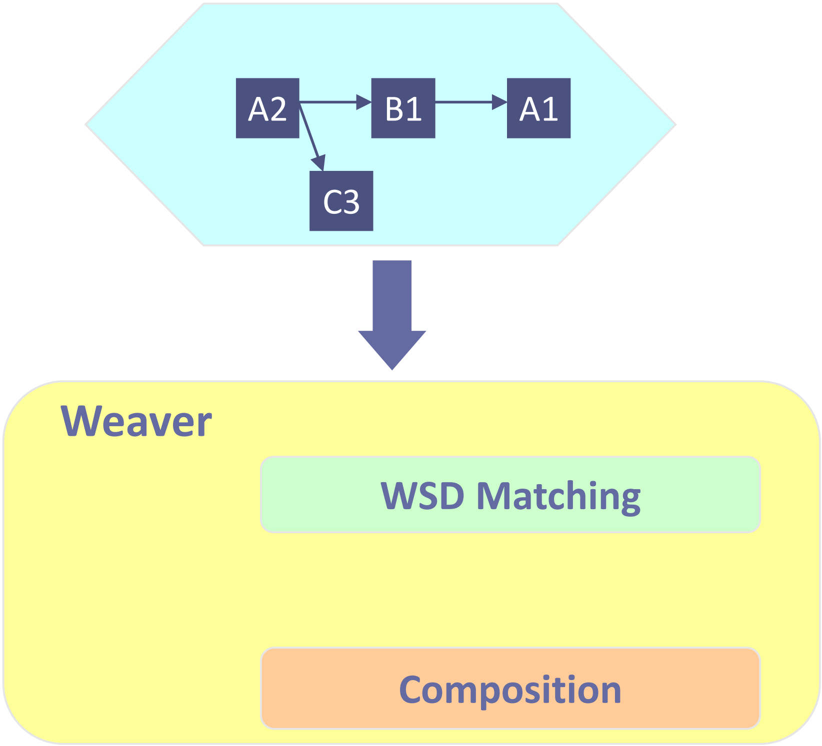
Our future works for semantic matching

- Structural with % similarities threshold,
- API matching

Second Challenge : Software Composition of multiple applications with shared Physical Devices

How to manage conflicts between Applications

Illustration with regular expression matching



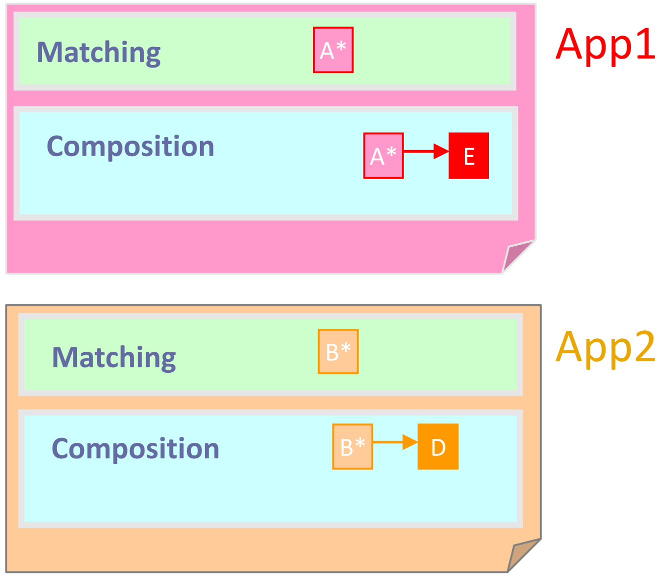


Illustration with regular expression matching

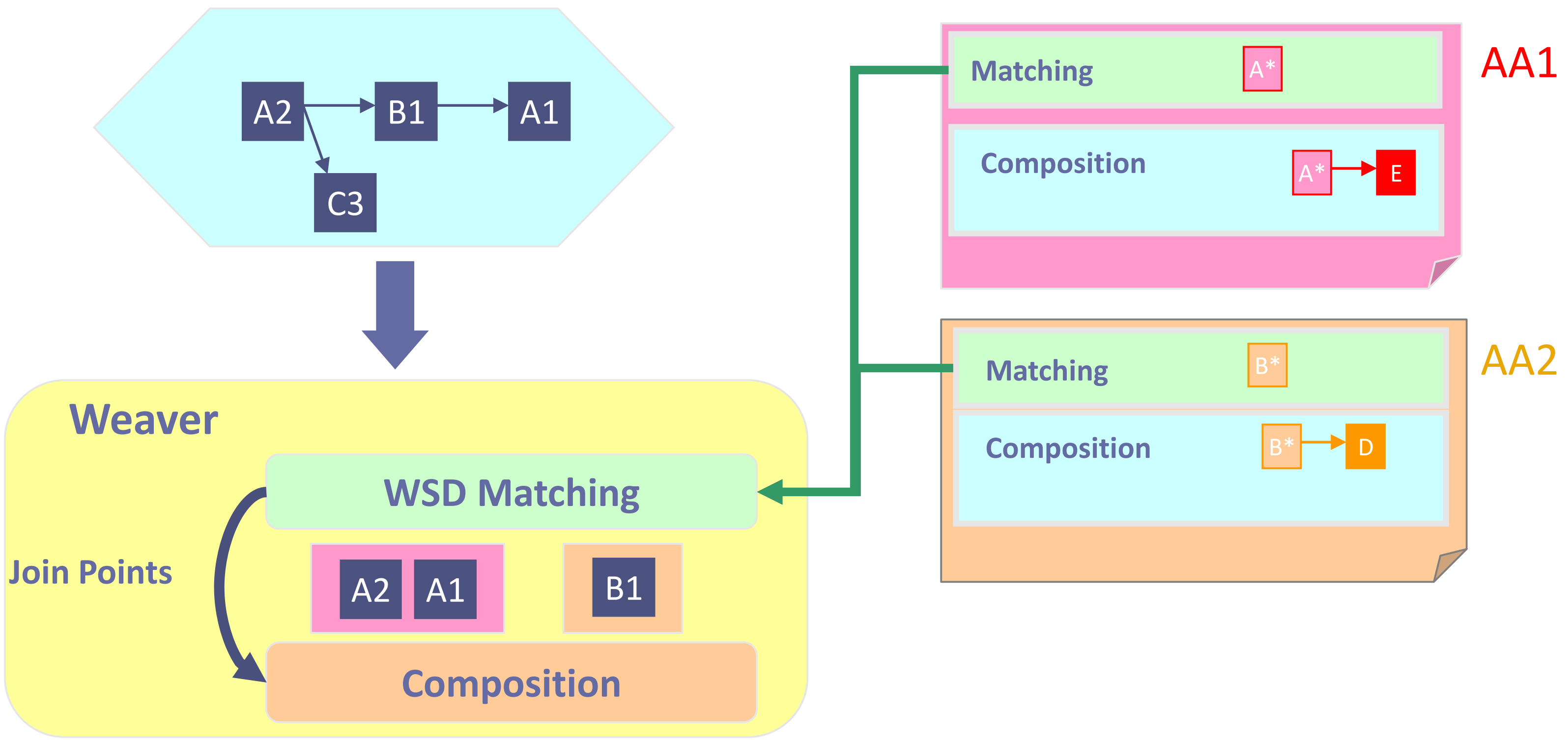
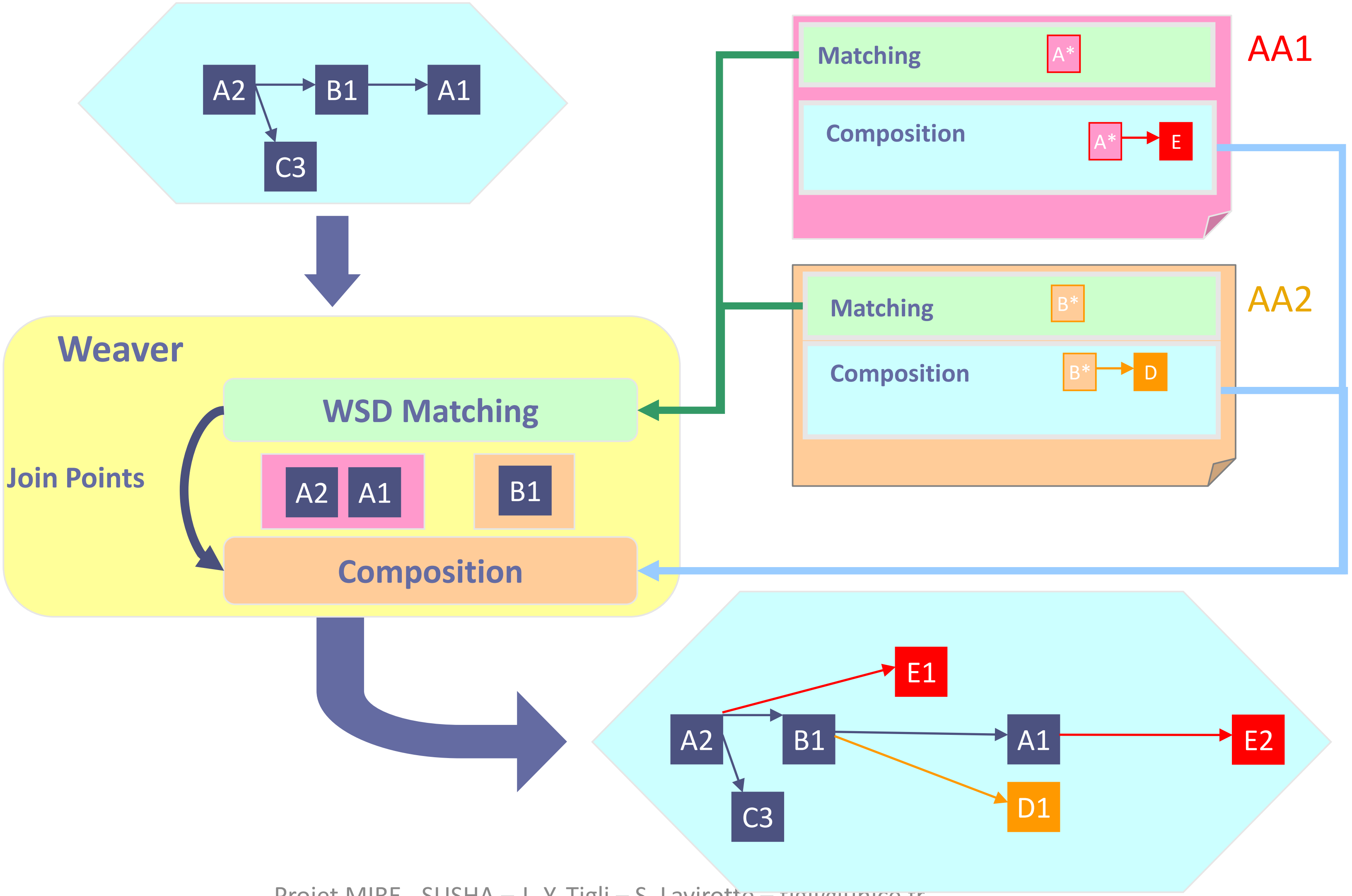
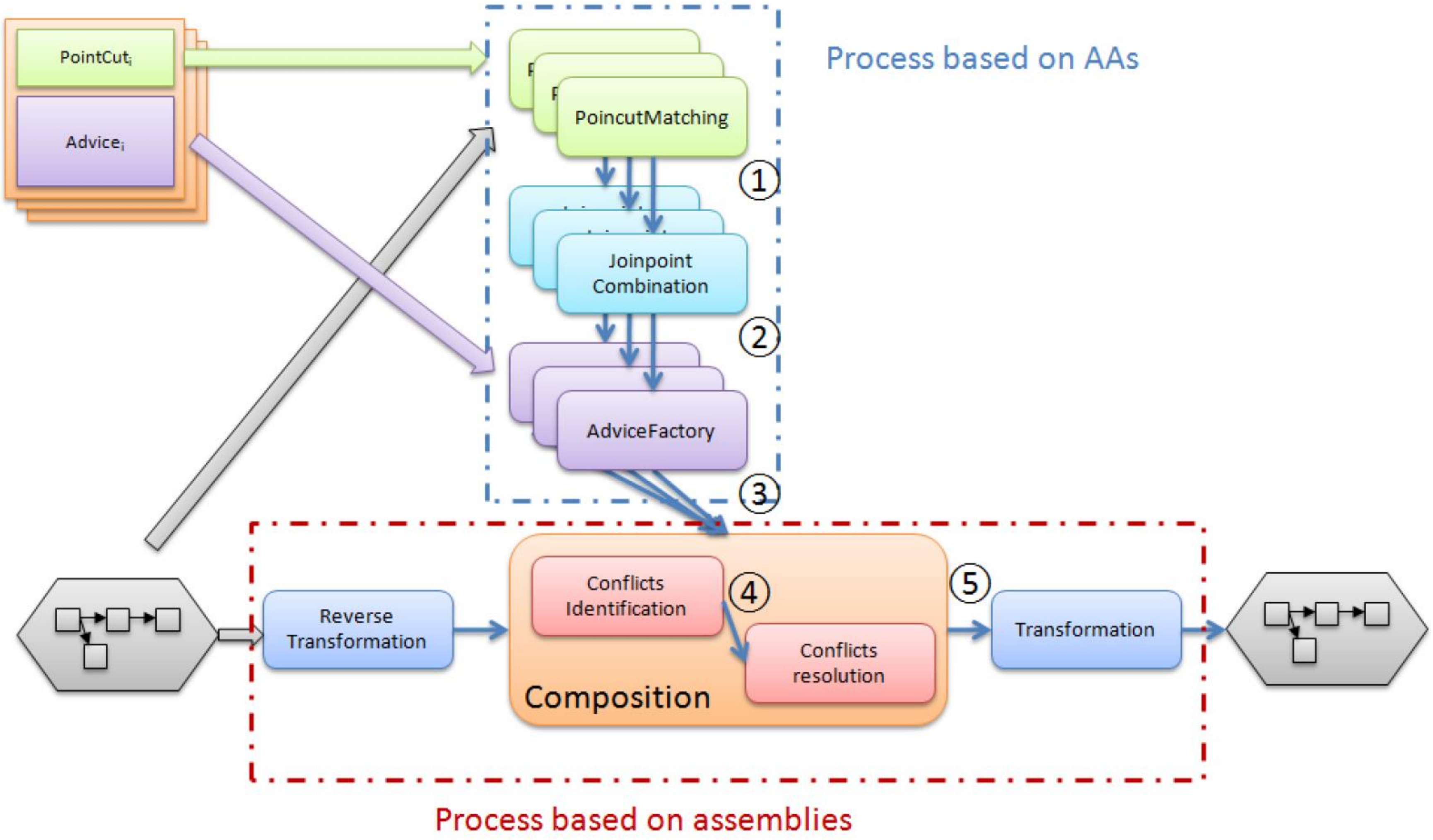


Illustration with regular expression matching



Complete Algorithm of one adaptation Cycle

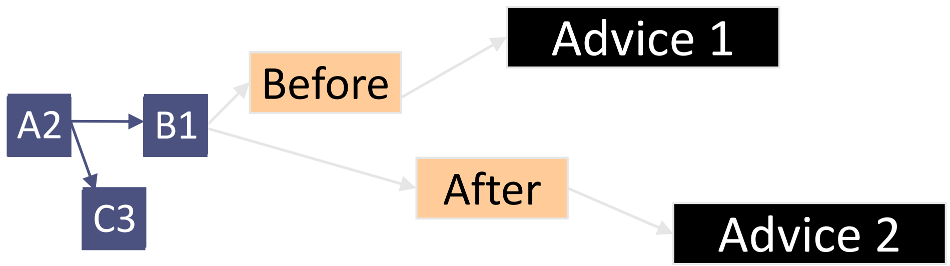


Two kinds of composition between applications around shared devices

✓ **Blackbox Advices :**

- **External composition between advices**

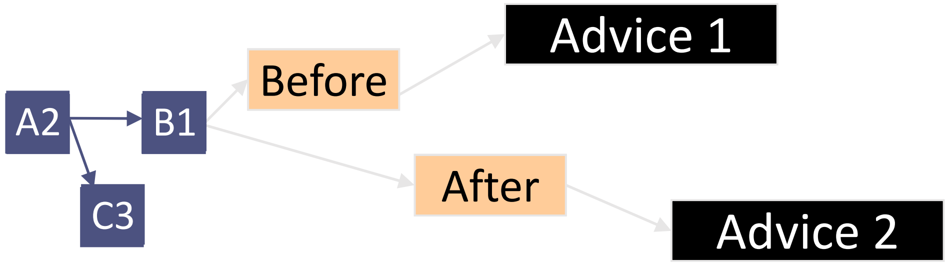
```
SCHEMA Ex (observed):
observed.^Out1 -> Bcomp.do;  CALL /* before
observed.^Out2 -> CALL ; Bcomp.do; /* after
```



Two kinds of composition between applications around shared devices

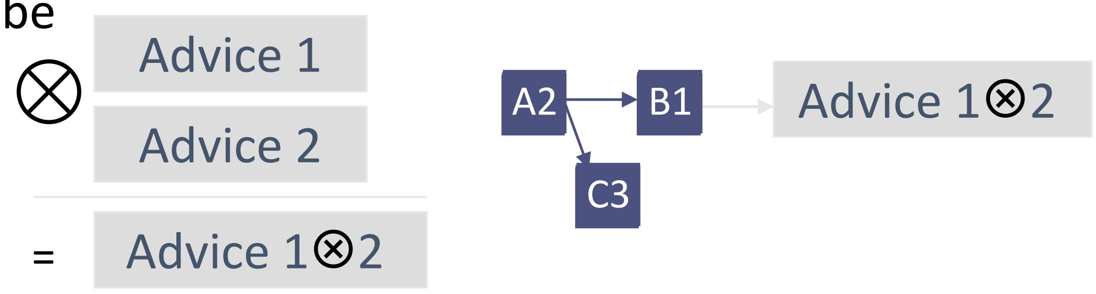
- Blackbox Advices :
  - External composition between advices

```
SCHEMA Ex (observed):
observed.^Out1 -> Bcomp.do;  CALL /* before
observed.^Out2 -> CALL ; Bcomp.do; /* after
```



- Greybox Advices :
  - We partly know the semantic of the advice
  - Advices are mergeabe

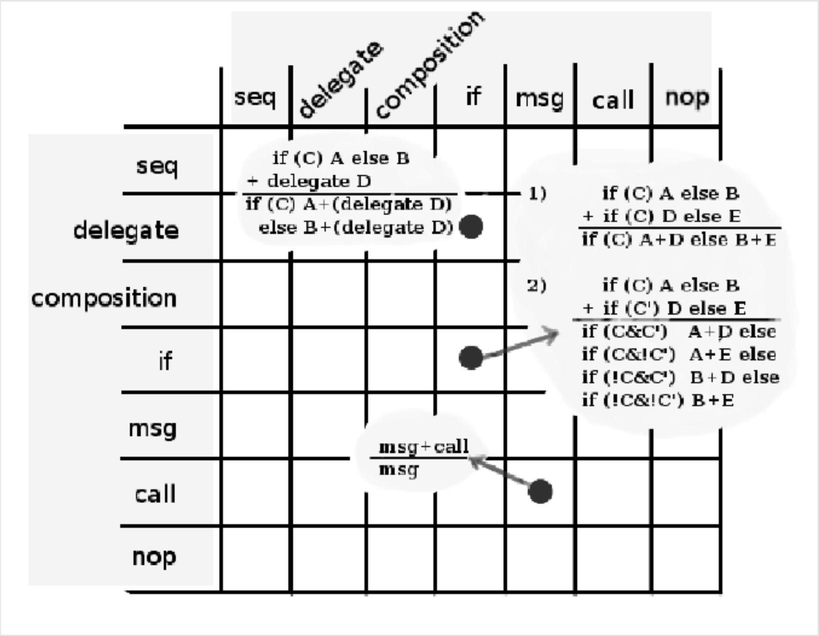
```
SCHEMA Ex (observed, timeout):
    observed.^Out ->
           ( IF ( timeout.Check ) CALL )
    timeout.Check ->
           ( timeout.Start ; CALL      )
```



Merge Logic

- Merge Logic,

Operators [Daniel Cheung, Ph. D. Thesis]



- Merge Properties

Properties Proof [Daniel Cheung, Ph. D. Thesis]

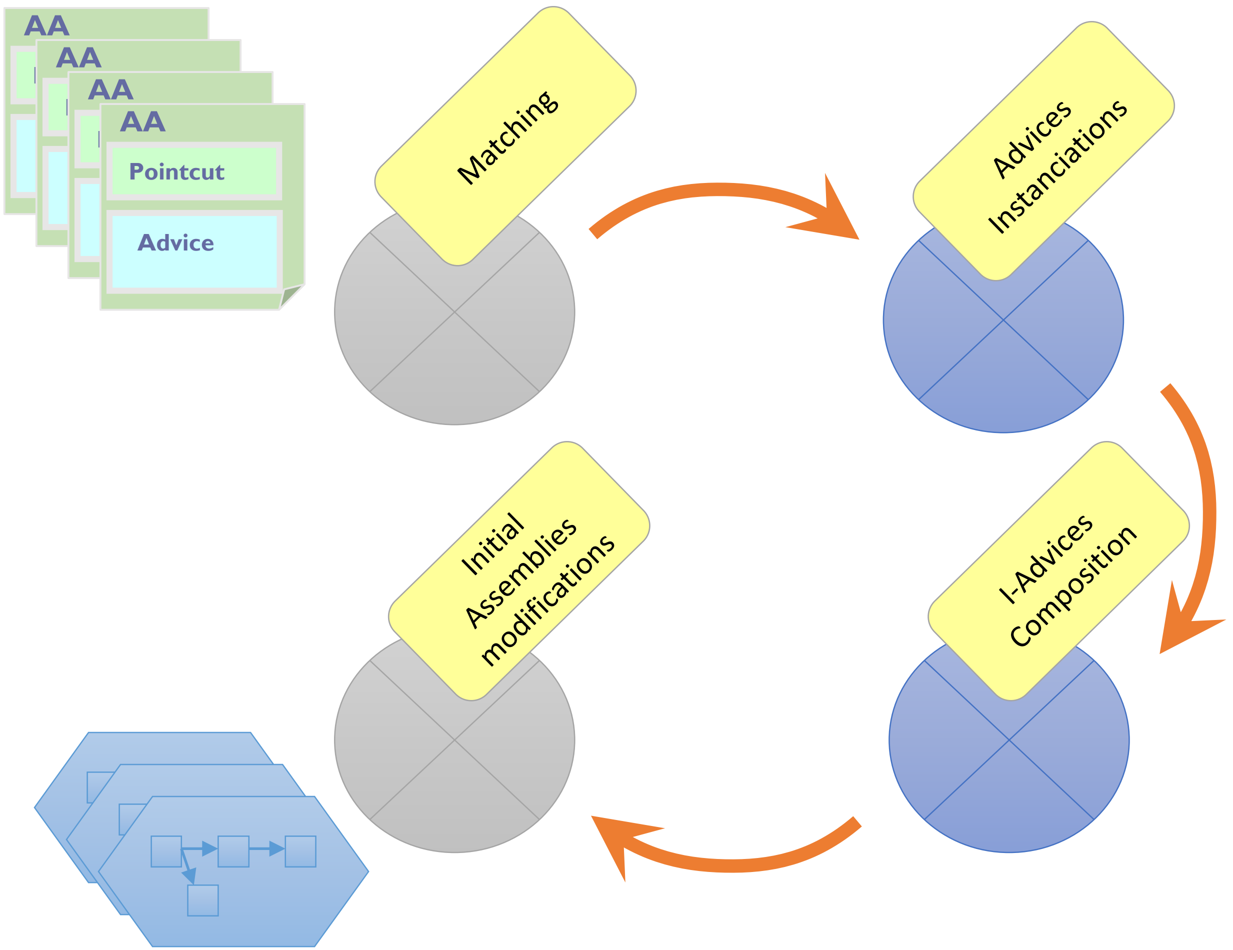
Commutativity : AA0 ⊗ AA1 = AA0 ⊗ AA1
Associativity : (AA0 ⊗ AA1) ⊗AA2 = AA0 ⊗ (AA1 ⊗ AA2)
Idempotence : AA0 ⊗ AA0 = AA0

Third Challenge : Is it reactive enough ?

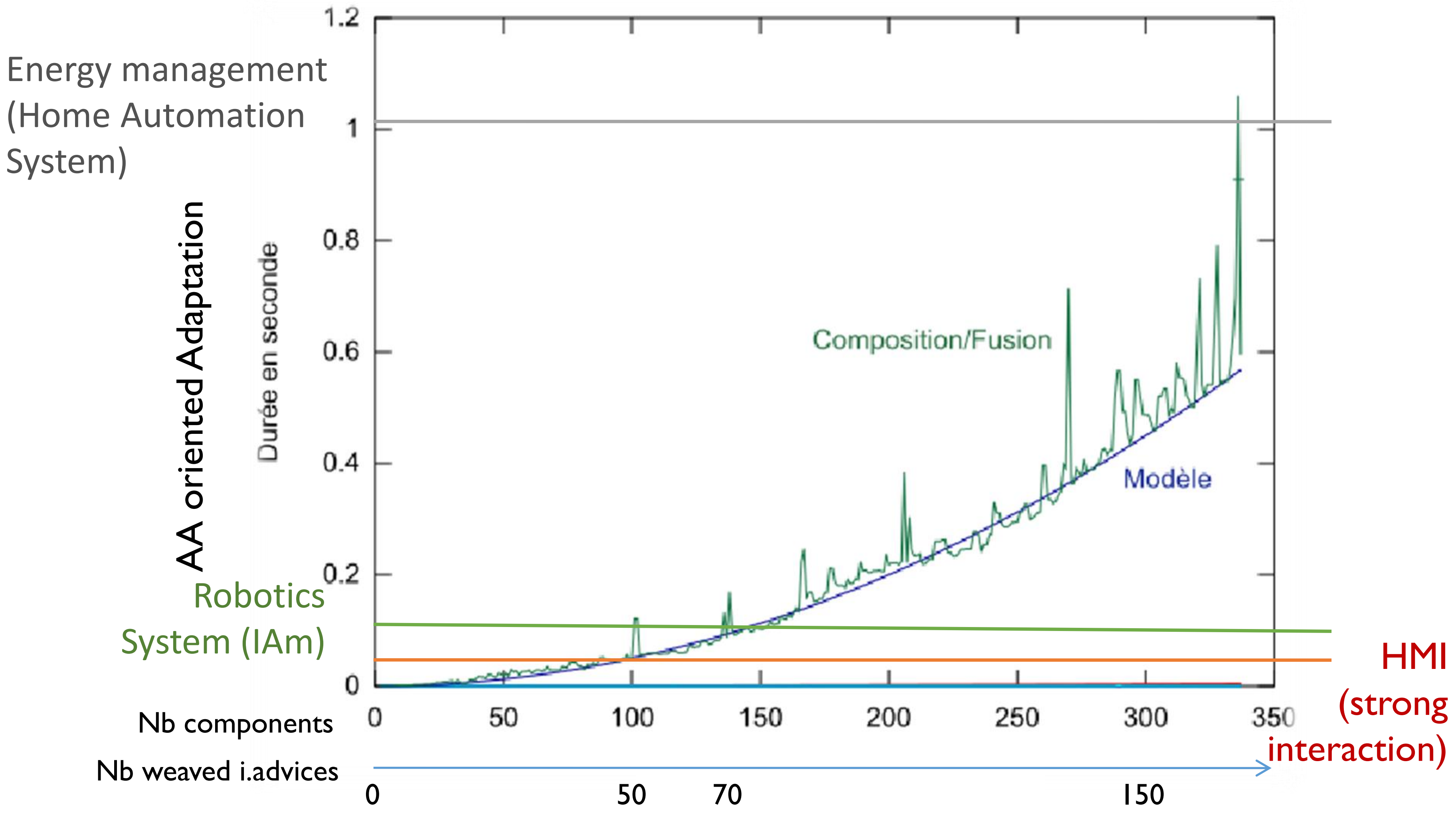
Response time model of one adaptation cycle

Seamless Services

Reactivity, response time and adaptation Cycle



Reactivity and Adaptation : Temporal Model and Experimental results



Application to seamless services

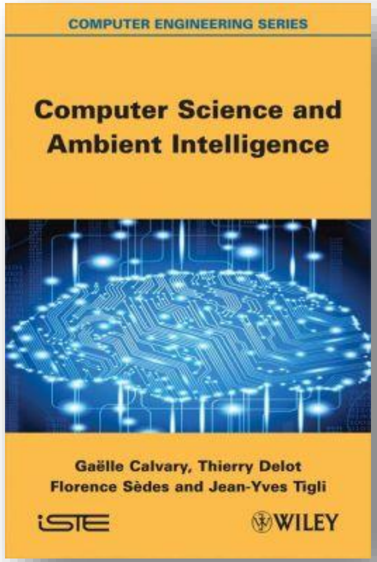
- ANR CONTINUUM Project
- http://continuum.unice.fr

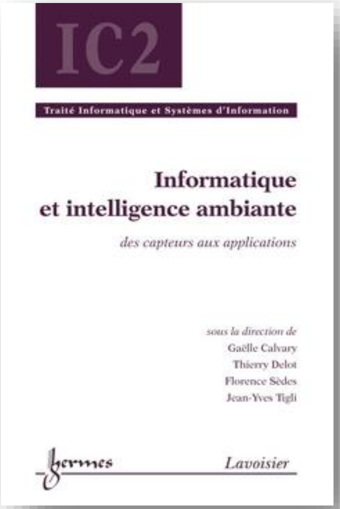
- See Video of IAm seamless applications for Hydran Man in the Water Industry

Some overview References

[2013] Gaëlle Calvary, Thierry Delot, Florence Sèdes, **Jean-Yves Tigli**, editors. “Computer Science and Ambient Intelligence” 335 pages, ISTE Ltd and Wiley & Sons Inc, March 2013, ISBN 978-1-84821-437-8

[2012] Gaëlle Calvary, Thierry Delot, Florence Sèdes, **Jean-Yves Tigli**. “Informatique et Intelligence Ambiante : des Capteurs aux Applications (Traité Informatique et Systèmes d'Information, IC2)” Hermes Science, July 2012, ISBN 2-7462-2981-1





One Ph. D. Current Work

G. Rocher

Advisor J.-Y. Tigli and N. Le Than

On the Behavioral Drift Estimation of Ubiquitous Computing Systems in Partially Known Environments

13th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services
November 28–December 1, 2016, Hiroshima, Japan

Gérald Rocher, Jean-Yves Tigli and Stéphane Lavirotte

NEW CHALLENGES, NEW IDEAS

sophia antipolis

dépasser les frontières

UNIVERSITÉ CÔTE D'AZUR

Introduction

- Calm technology paradigm (M.Weiser, 1995)…
  - Technology *disappears* from view,
  - Ubiquitous applications interact *seamlessly* with users and their surroundings
  - …

Introduction

- **Calm technology paradigm (M.Weiser, 1995)…**
  - Technology *disappears* from view,
  - Ubiquitous applications interact *seamlessly* with users and their surroundings
  - …

✓ Proliferation of connected objects in our surroundings,
✓ Ubiquitous applications operate in the physical environment via effectors & sensors.

Introduction

- Calm technology paradigm (M.Weiser, 1995)…
  - Technology *disappears* from view,
  - Ubiquitous applications interact *seamlessly* with users and their surroundings,
  - …

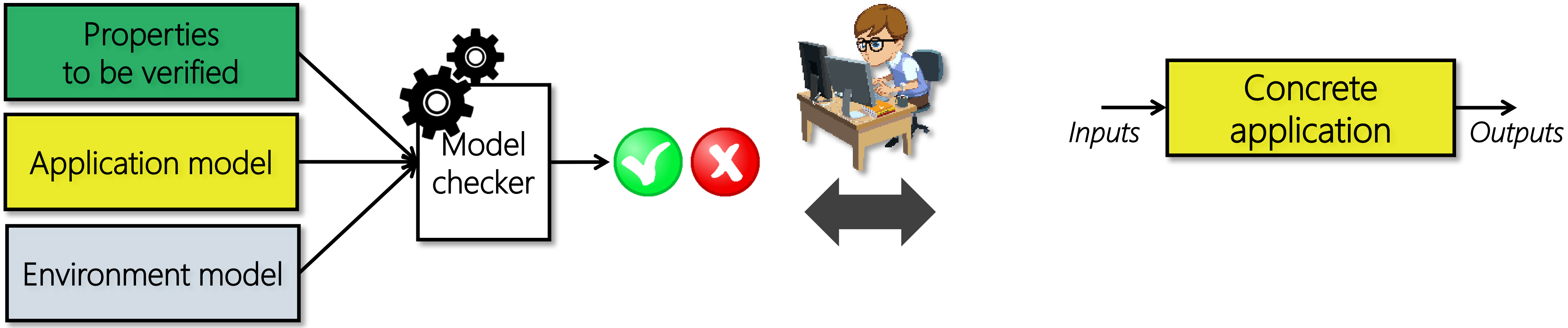
- … is not yet a reality
  - ✓ Proliferation of connected objects in our surroundings,
  - ✓ Ubiquitous applications operate in the physical environment via effectors & Sensors.
  - ✗ Responsibility of the actions to be undertaken delegated to the users.

Introduction

- Calm technology paradigm (M.Weiser, 1995)…
  - Technology *disappears* from view,
  - Ubiquitous applications interact *seamlessly* with users and their surroundings,
  - …
- … is not yet a reality
  - ✓ Proliferation of connected objects in our surroundings,
  - ✓ Ubiquitous applications operate in the physical environment via effectors & Sensors.
  - ✗ Responsibility of the actions to be undertaken delegated to the users.

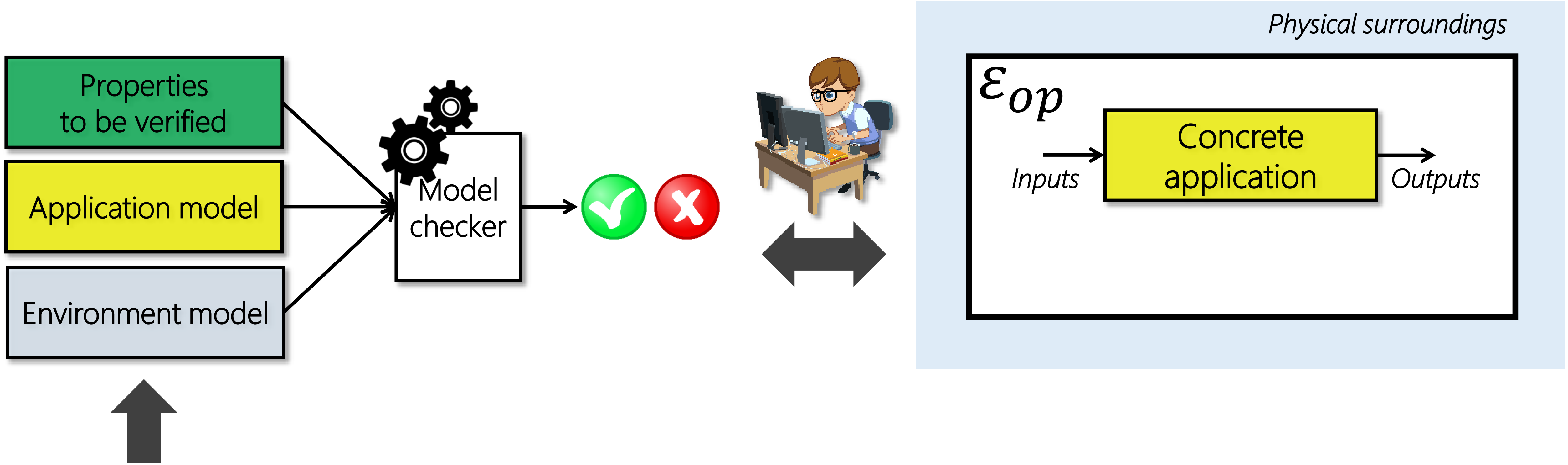
Problem statement

- Model Driven Engineering (MDE) technics used to *verify/predict* applications behavior.



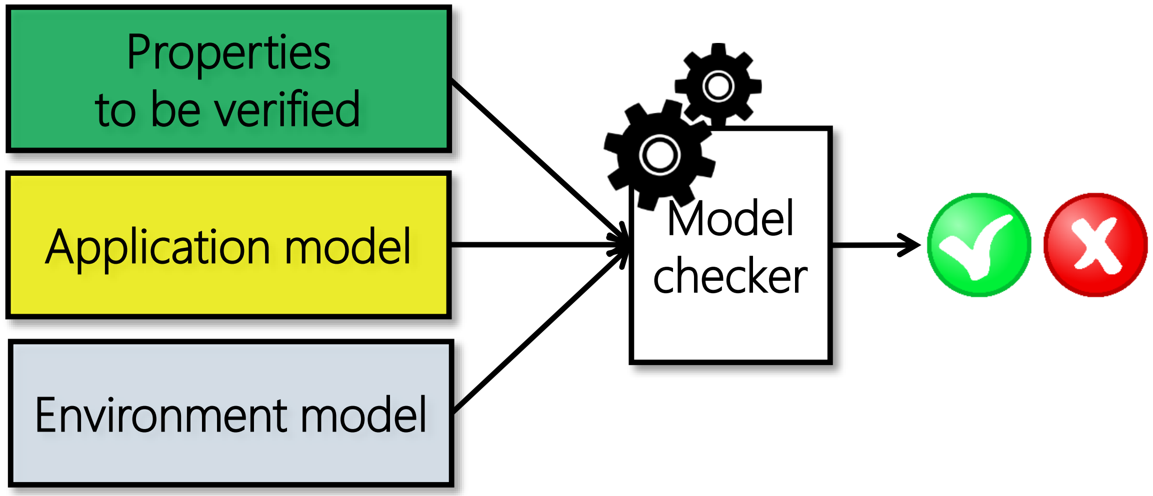
Problem statement

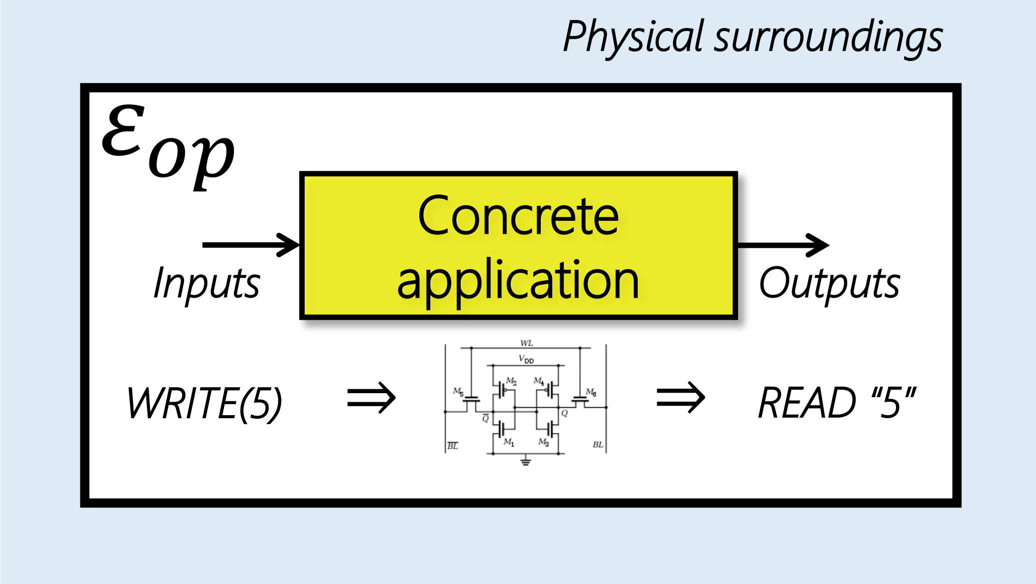
- Assume ***deterministic/controlled*** behavior while operating in the environment



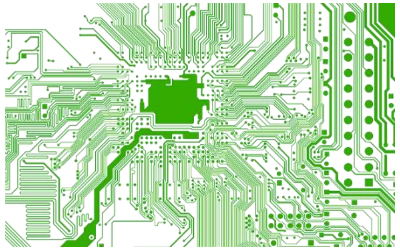
Problem statement

- Assume *deterministic/controlled* behavior while operating in the environment



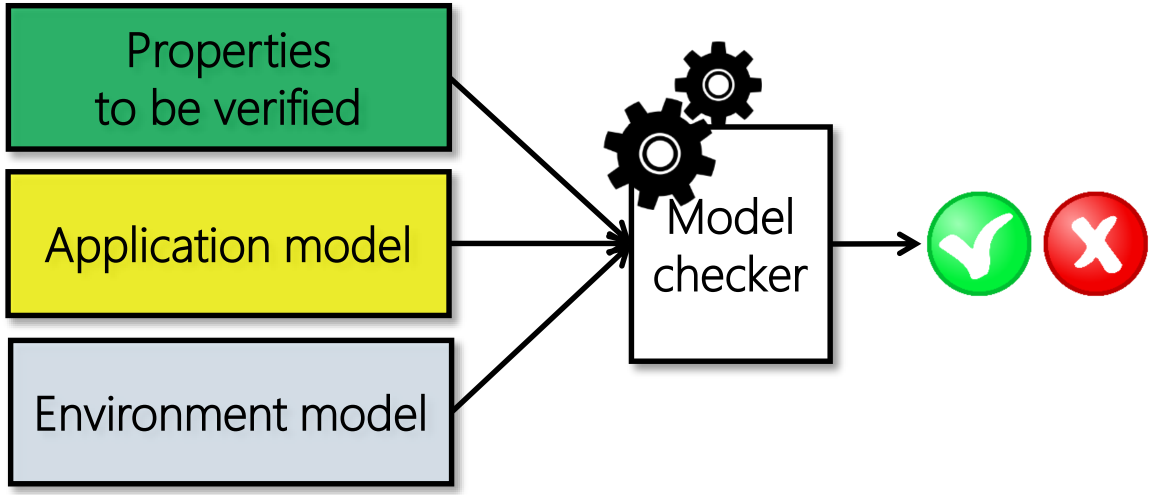


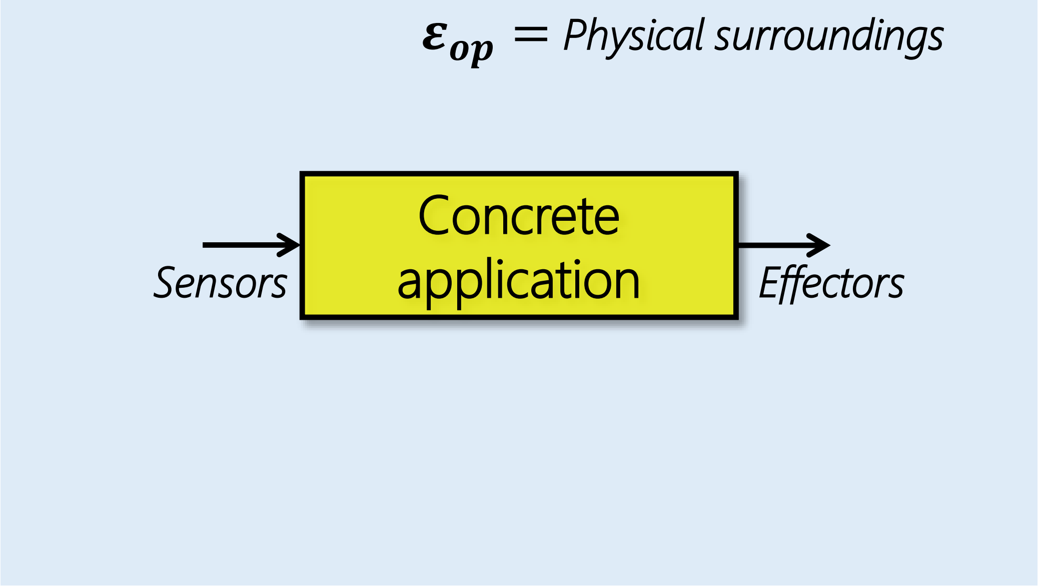
What You Set Is What You Get!



Problem statement

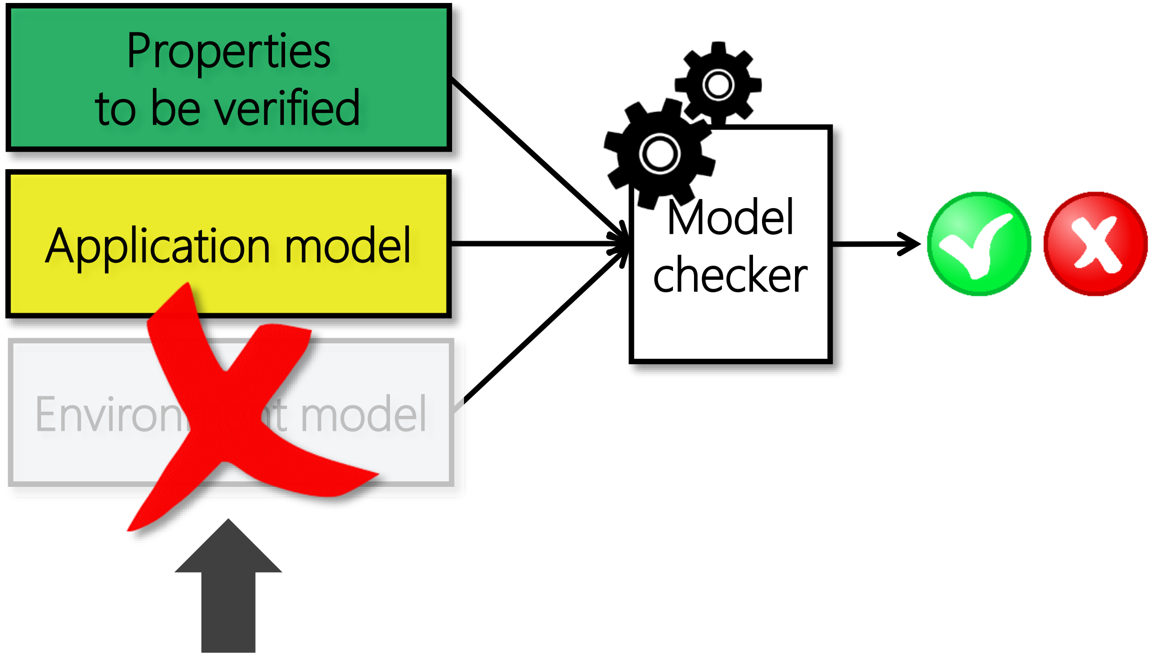
- However, the physical environment is open and subject to *uncertainties*...

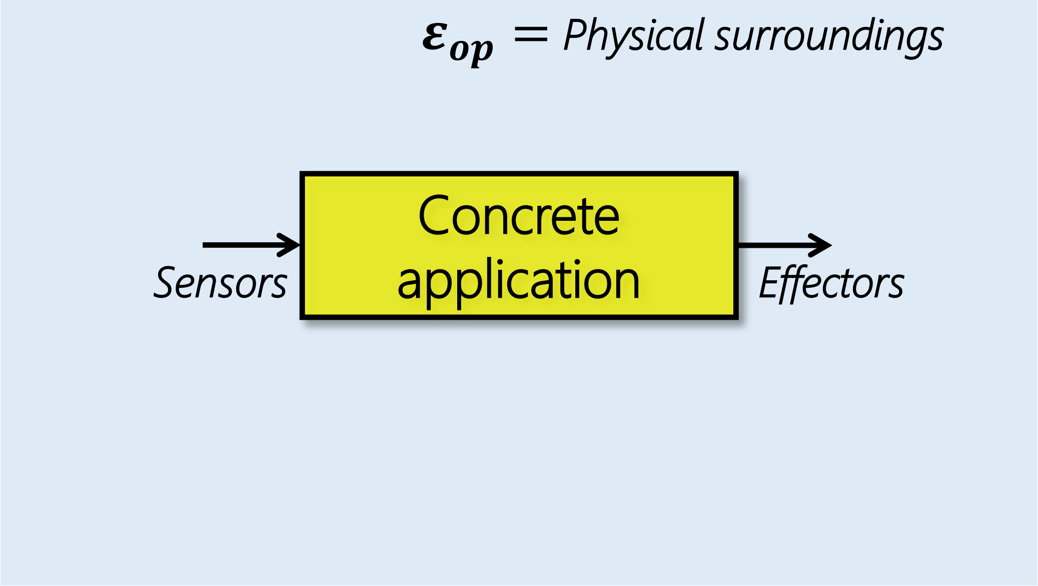




Problem statement

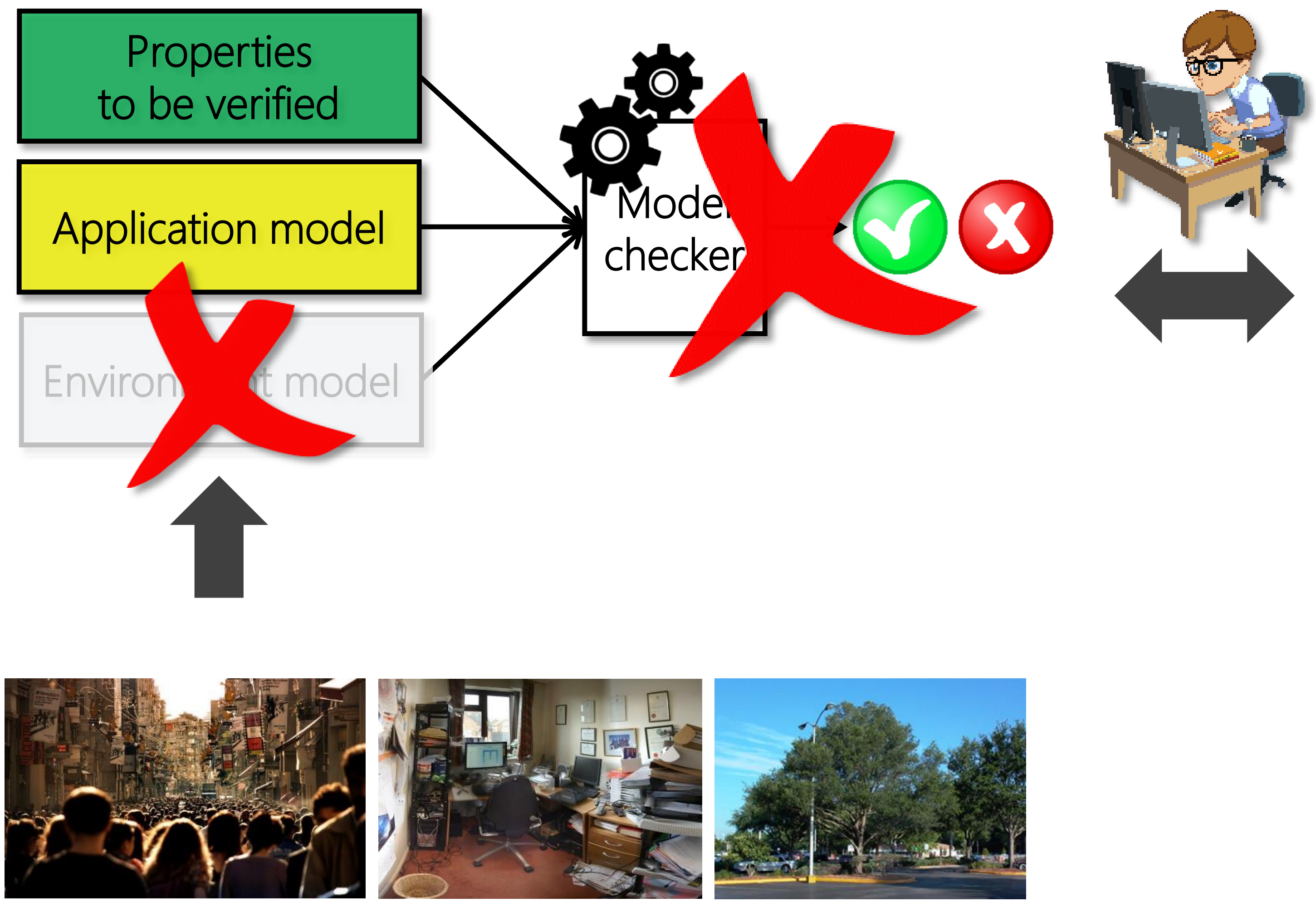
- … that cannot be accurately and entirely modeled.

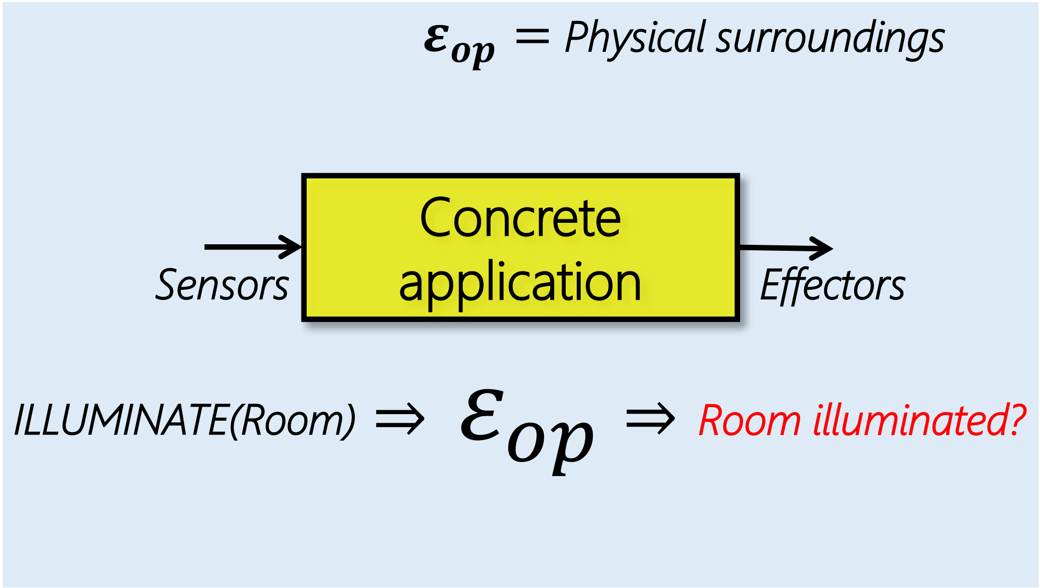




Problem statement

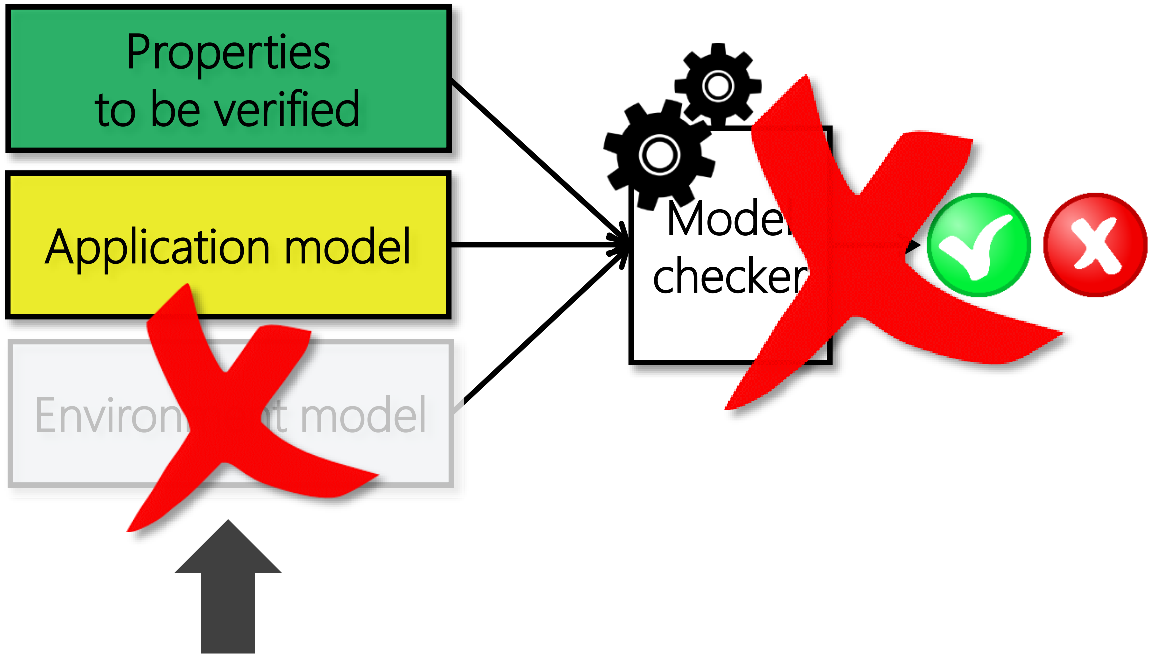
- In this context MDE technics are not well suited to verify/predict behavior.

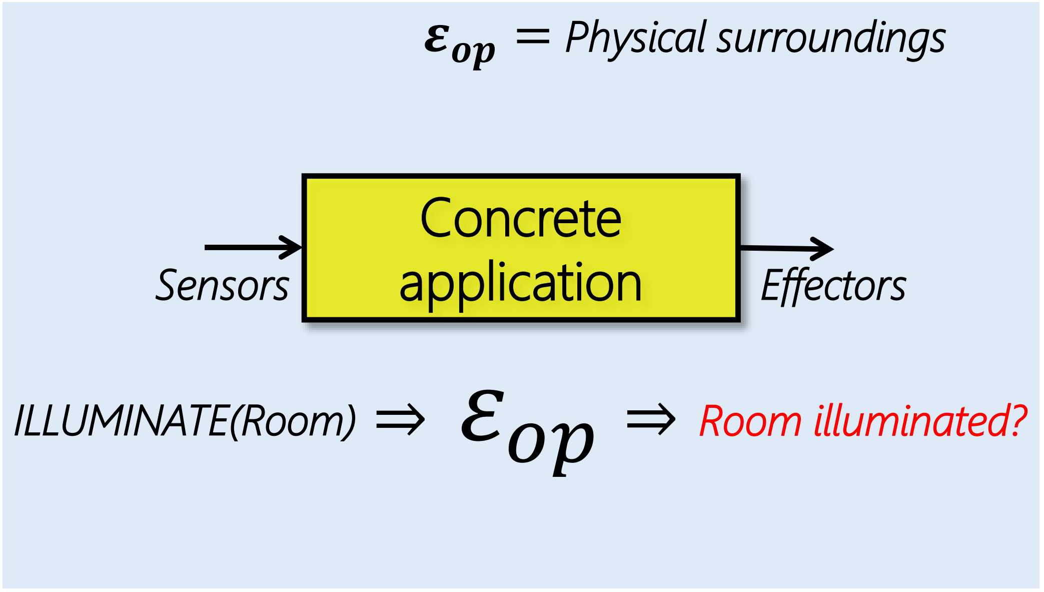




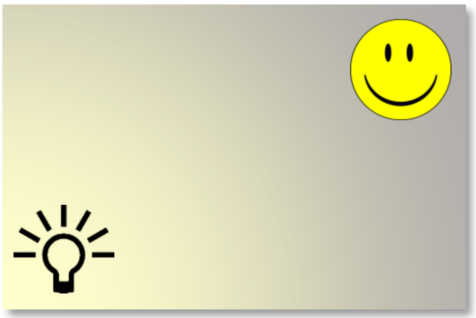
Problem statement

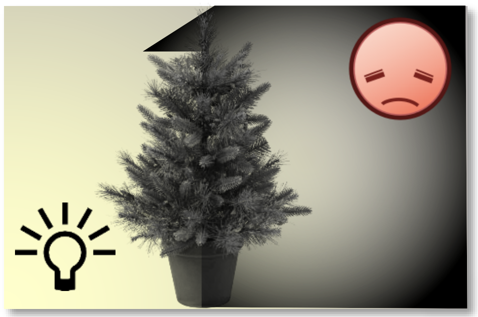
- In this context MDE technics are not well suited.





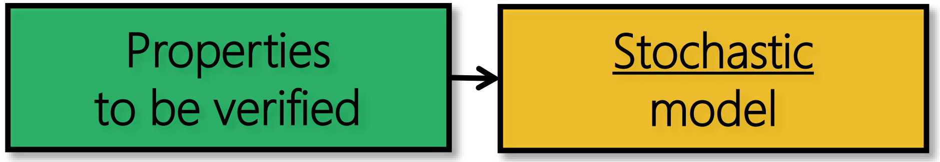
What You Set Is <u>NOT</u> What You Get!

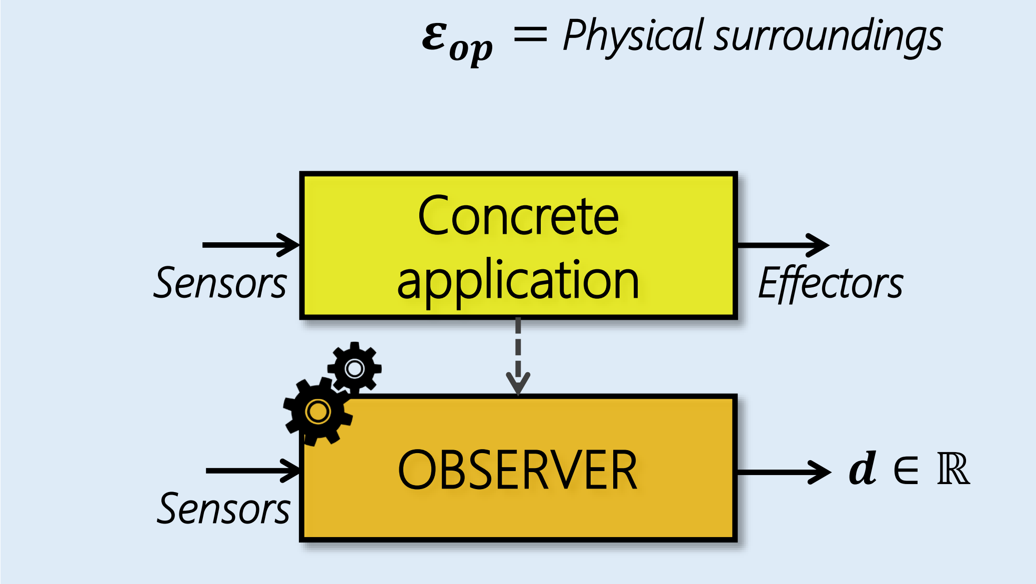




Proposed approach

- We appeal on the control theory and the notion of state observer…

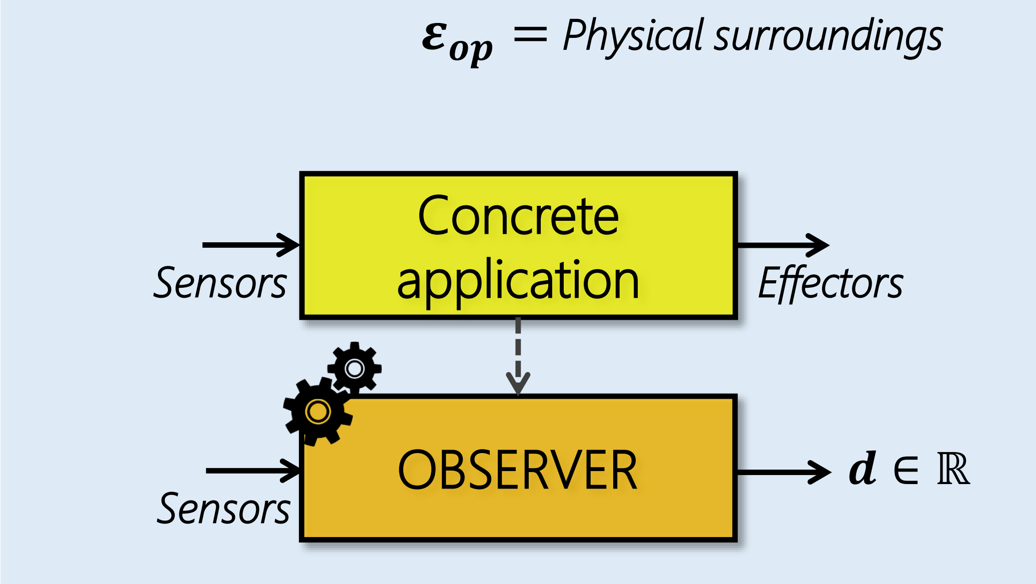




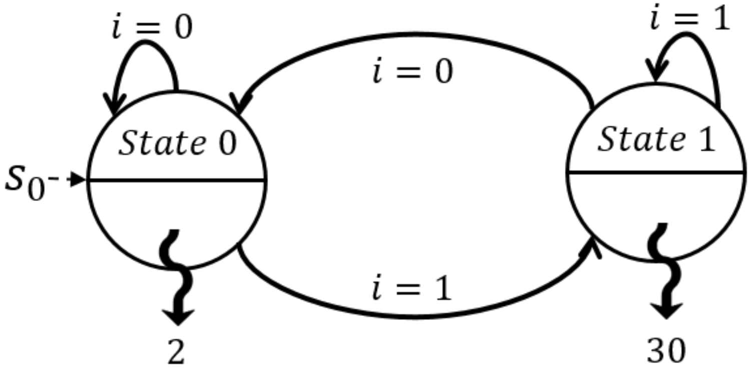
Proposed approach

- Ideally, the behavior of an application is supposed to be deterministic…

Properties to be verified → Stochastic model



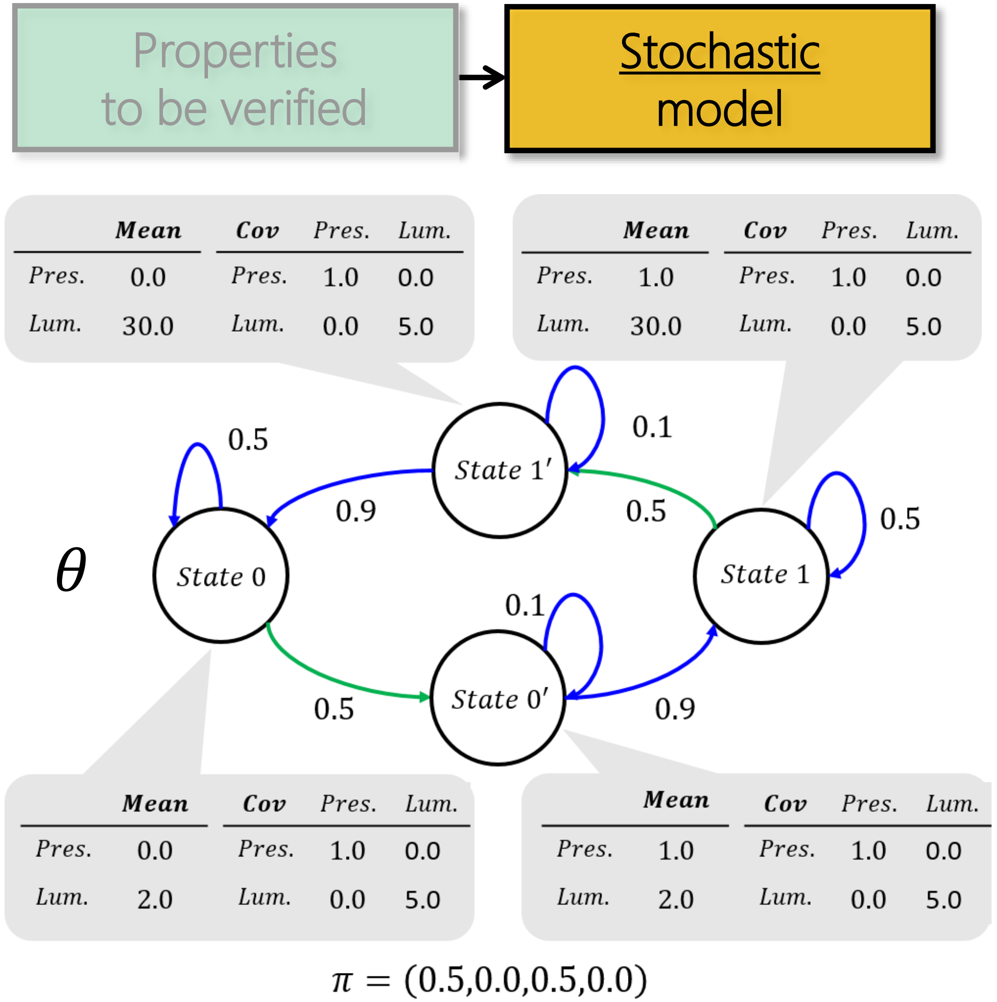
Input i (presence sensor)	Low luminosity (state 0)	High luminosity (state 1)
No presence detected	Conform	Non – conform
Presence detected	Non – conform	Conform

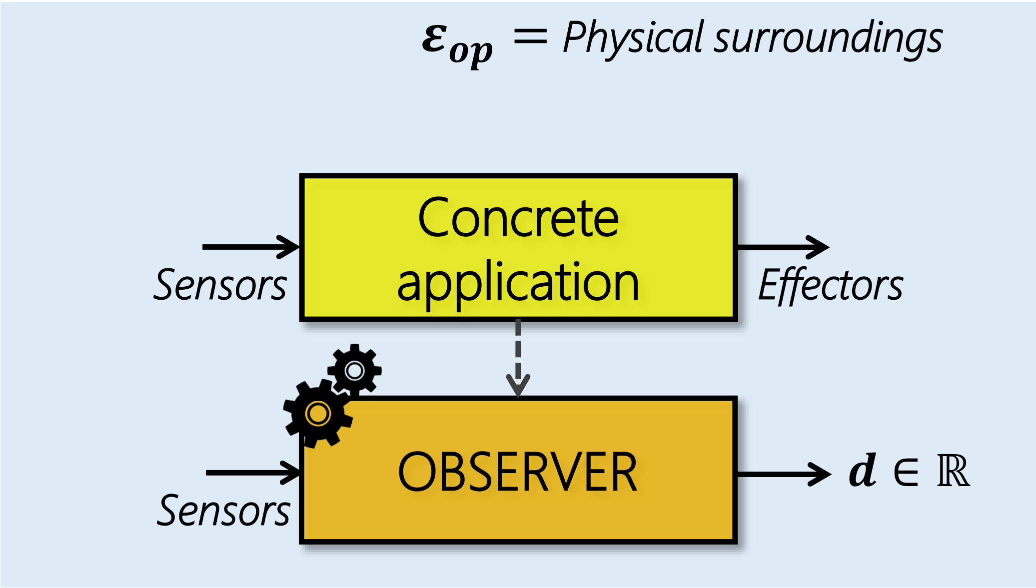


⇒ Deterministic Finite State Machine
⇒ Moore DFSA

Proposed approach

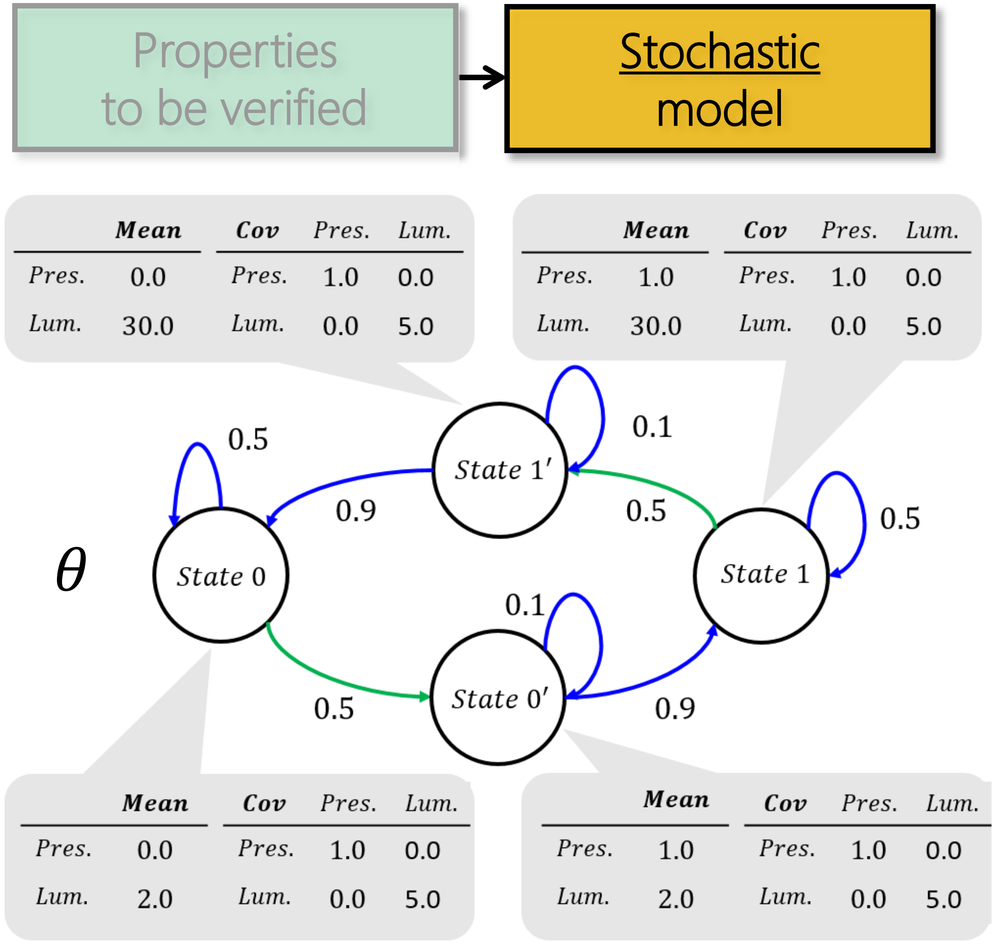
- … but more realistically, is subject to uncertainties leading to introduce <u>probabilities</u>.





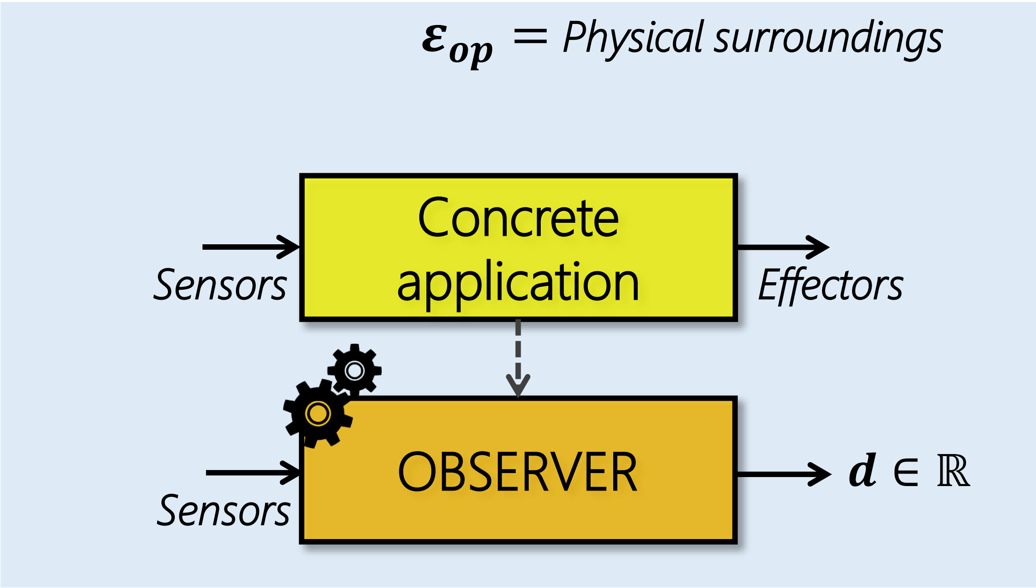
Proposed approach

- … but more realistically, is subject to uncertainties leading to introduce <u>probabilities</u>.



$\pi = (0.5, 0.0, 0.5, 0.0)$

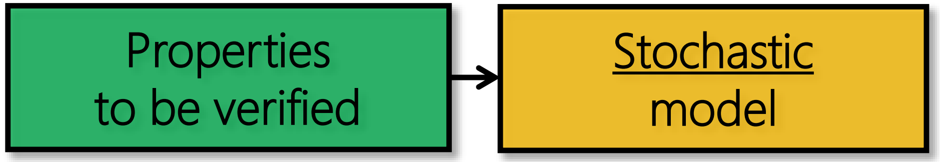
$\Rightarrow \mathbb{P}(y_1^T | \theta) \in \mathbb{R}$

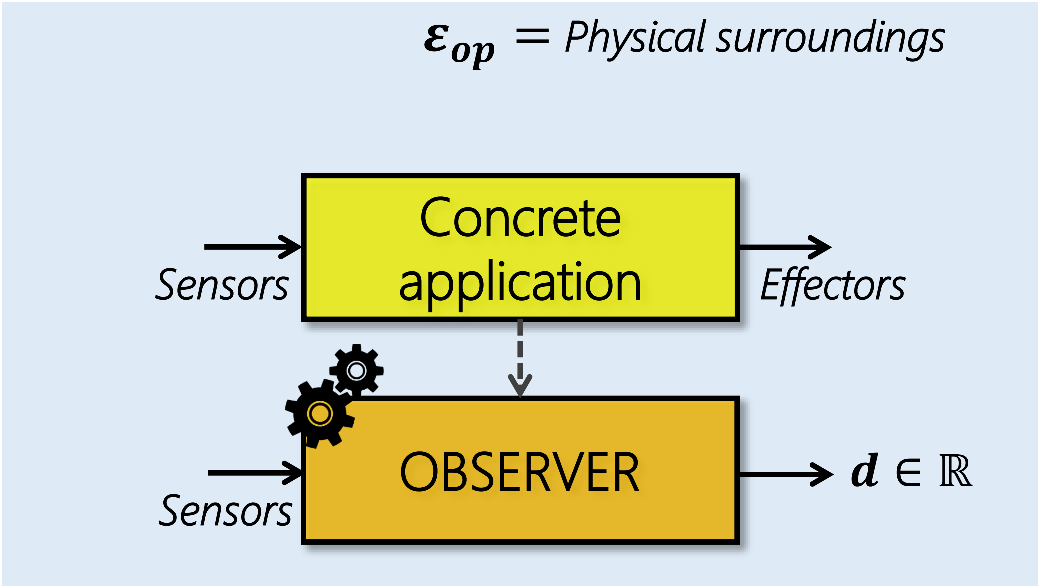


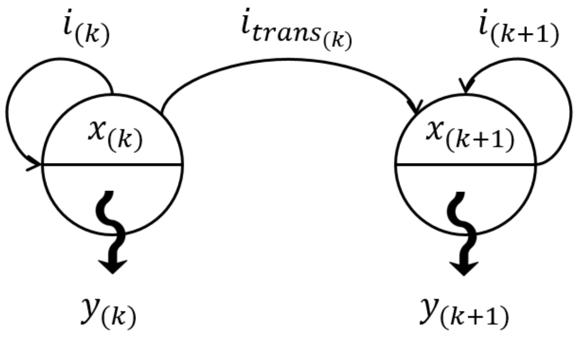
⇒ Continuous Density Hidden Markov Model
⇒ Parameters set by the user or learnt.

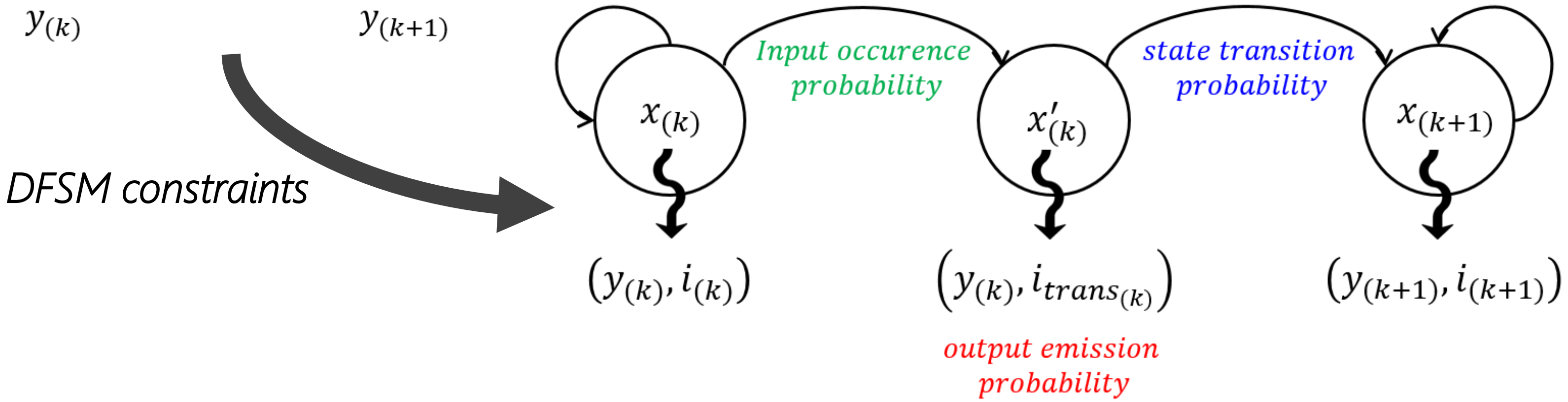
From deterministic to Probabilistic model

- Introducing input-occurrence and state-transition probabilities…

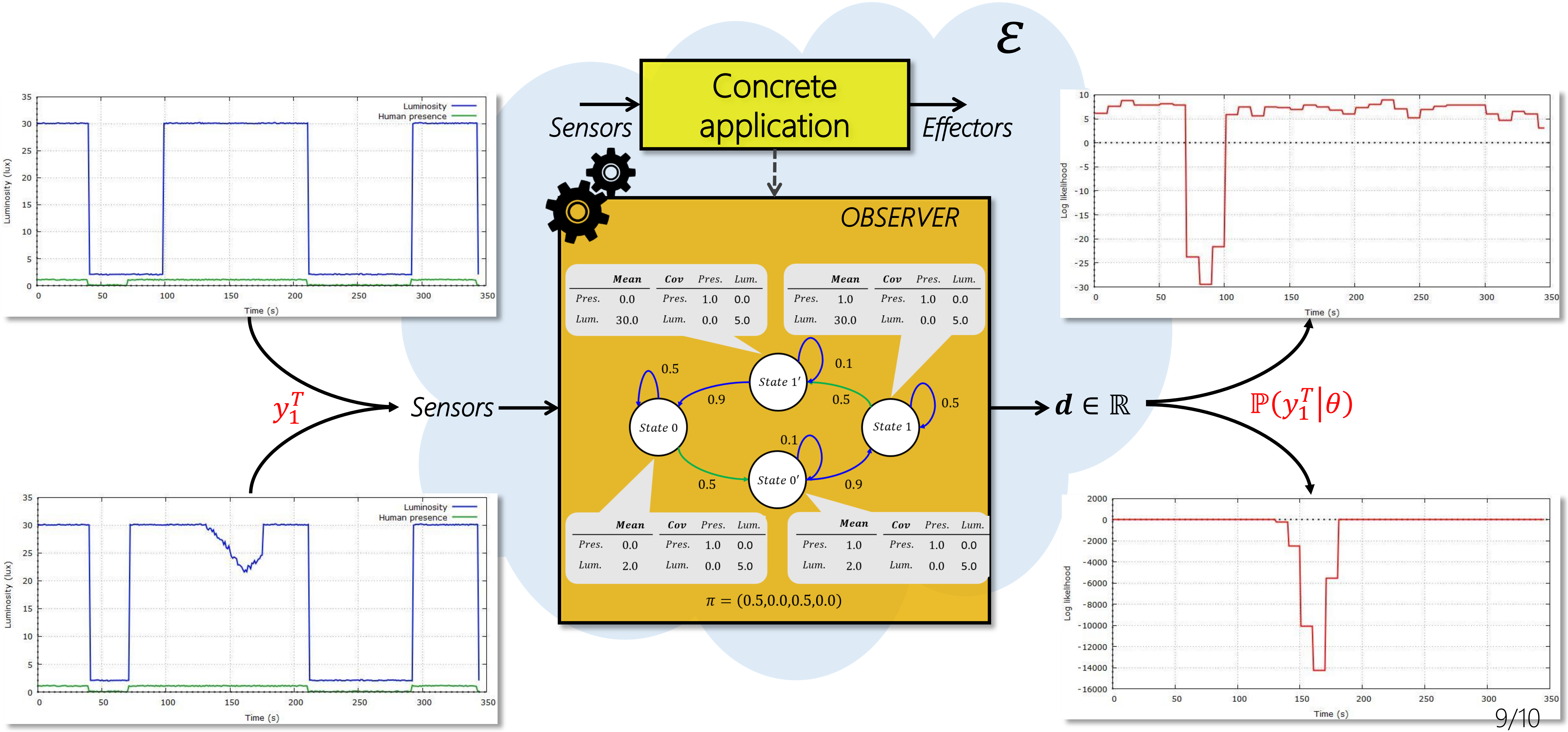








Experimentation results



Conclusion

- Calm technology not yet a reality
  - The physical environment as an operational environment is not reliable,
  - Classical MDE technics as a means to verify/predict applications behavior are no longer adequate.

Conclusion

- Calm technology is not yet a reality
  - The physical environment as an operational environment is not reliable,
  - Classical MDE technics as a means to verify/predict applications behavior are no longer adequate.

- Contribution
  - Provision ubiquitous systems with a run-time estimation ($\in \mathbb{R}$) of the applications behavioral conformity …
  - … that could be used in closed-loop systems (self-adaptiveness)

Conclusion

- Calm technology is not yet a reality
  - The physical environment as an operational environment is not reliable,
  - Classical MDE technics as a means to verify/predict applications behavior are no longer adequate.
- Contribution
  - Provision ubiquitous systems with a run-time estimation ($\in \mathbb{R}$) of the applications behavioral conformity …
  - … that could be used in closed-loop systems (self-adaptiveness).
- Future work
  - Scaling: address the state “explosion” problem,
  - Temporal constraints (physical process with inertia).

On the Behavioral Drift Estimation of Ubiquitous Computing Systems in Partially Known Environments

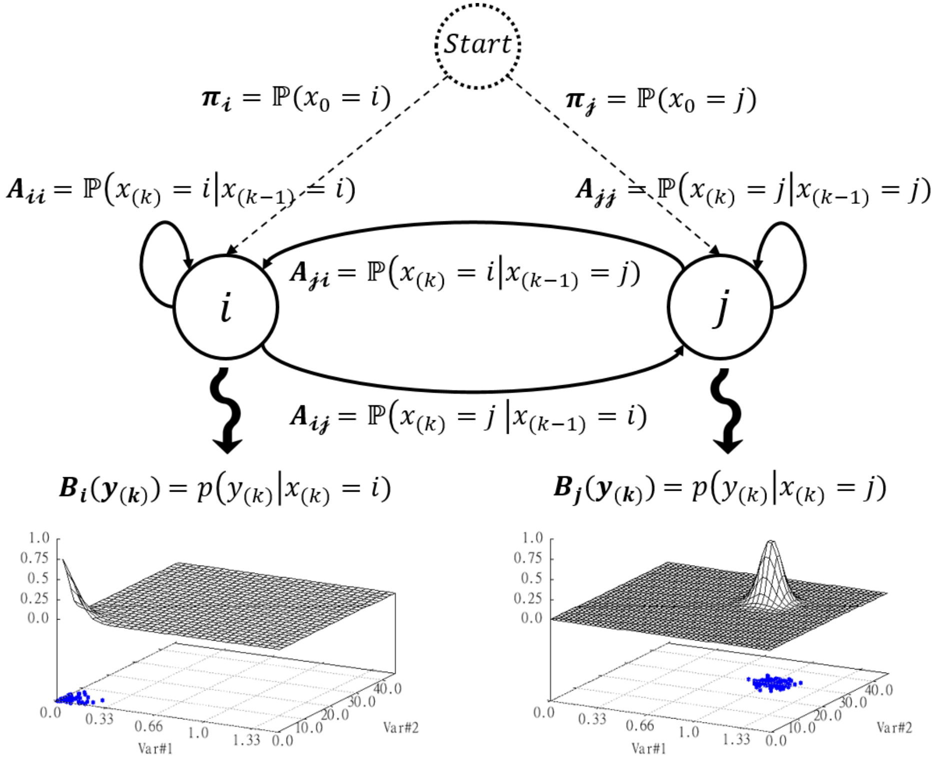
13th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services
November 28–December 1, 2016, Hiroshima, Japan

Thank you for your attention!

Gérald Rocher, Jean-Yves Tigli and Stéphane Lavirotte

Hidden Markov Model (HMM)

- $\theta = (A, B, \pi)$:
  - A, the $N \times N$ state-transition probability matrix (where N is the number of hidden states),
  - B, observation probability density functions (pdf) matrix,
  - π, the initial state distribution vector.

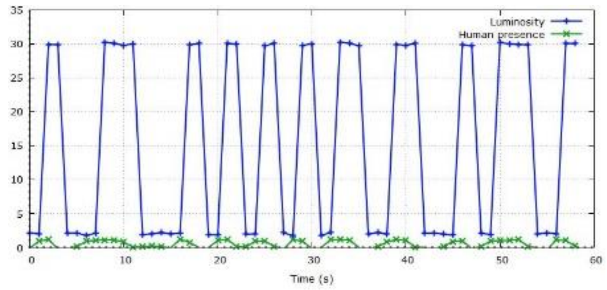


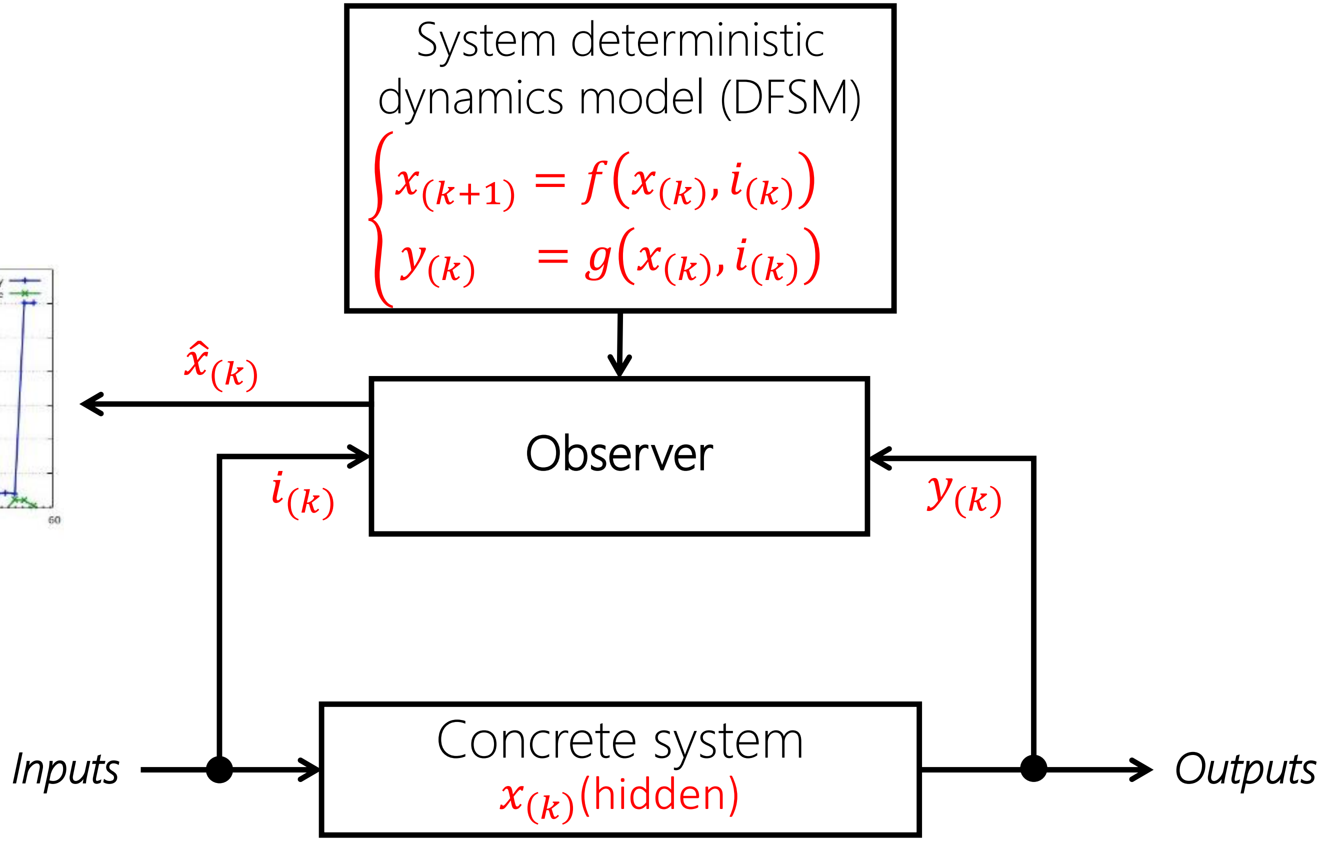
- Canonical problems
  1) **Given the model $\boldsymbol{\theta}$, compute the probability of an output sequence $\boldsymbol{y_1^T}$,**
  2) Given the model $\boldsymbol{\theta}$ and an output sequence $\boldsymbol{y_1^T}$, compute the most probable hidden state sequence $\boldsymbol{\hat{x}_1^T}$,
  3) Given an output sequence $\boldsymbol{y_1^T}$, compute the parameters of the model $\boldsymbol{\theta}$.

- Probabilities computation
  - State-transitions/initial distribution probabilities from DFSM constraints,
  - Considering multivariate normal density functions:
    - Mean values from DFSM emission values,
    - Variance/covariance values from users or learnt.

Deterministic state observer

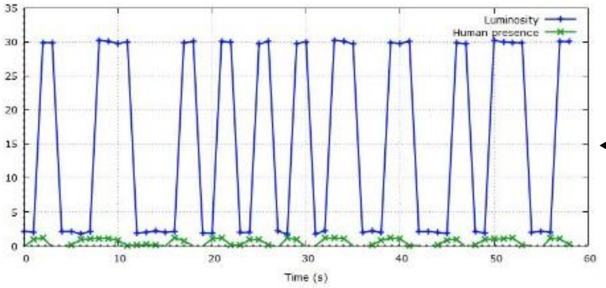
- Estimation of the state $\hat{x}$ of the system (hidden)
- <u>Deterministic system dynamics:</u>

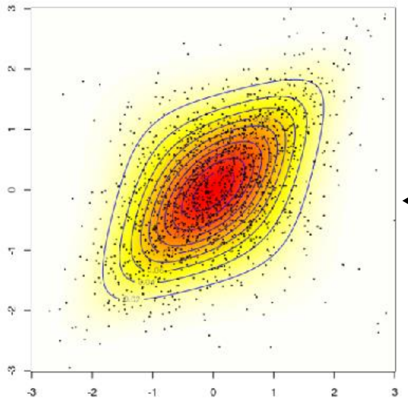




Probabilistic state observer

State sequence decoding ($\hat{x}_1^T$)





Probability of the observed sequence:
$\mathbb{P}(y_1^T|\theta)$

System deterministic model
(Moore DFSM)

$$\begin{cases} x_{(k+1)} = f(x_{(k)}, i_{(k)}) \\ y_{(k)} \quad = g(x_{(k)}) \end{cases}$$

Probabilistic model θ
(HMM)

$$\begin{cases} x_{(k+1)} \Rightarrow \mathbb{P}(x_{(k+1)}|x_{(k)}) \\ y_{(k)} \quad \Rightarrow \mathbb{P}(y_{(k)}|x_{(k)}) \end{cases}$$

$y_1^T = y_1, \dots, y_T$

$\mathcal{E}$

Sensing
Inputs

Concrete application

$x_{(k)}$(hidden)

Effecting
Outputs