Where do behaviour models come from?

Sebastian Uchitel

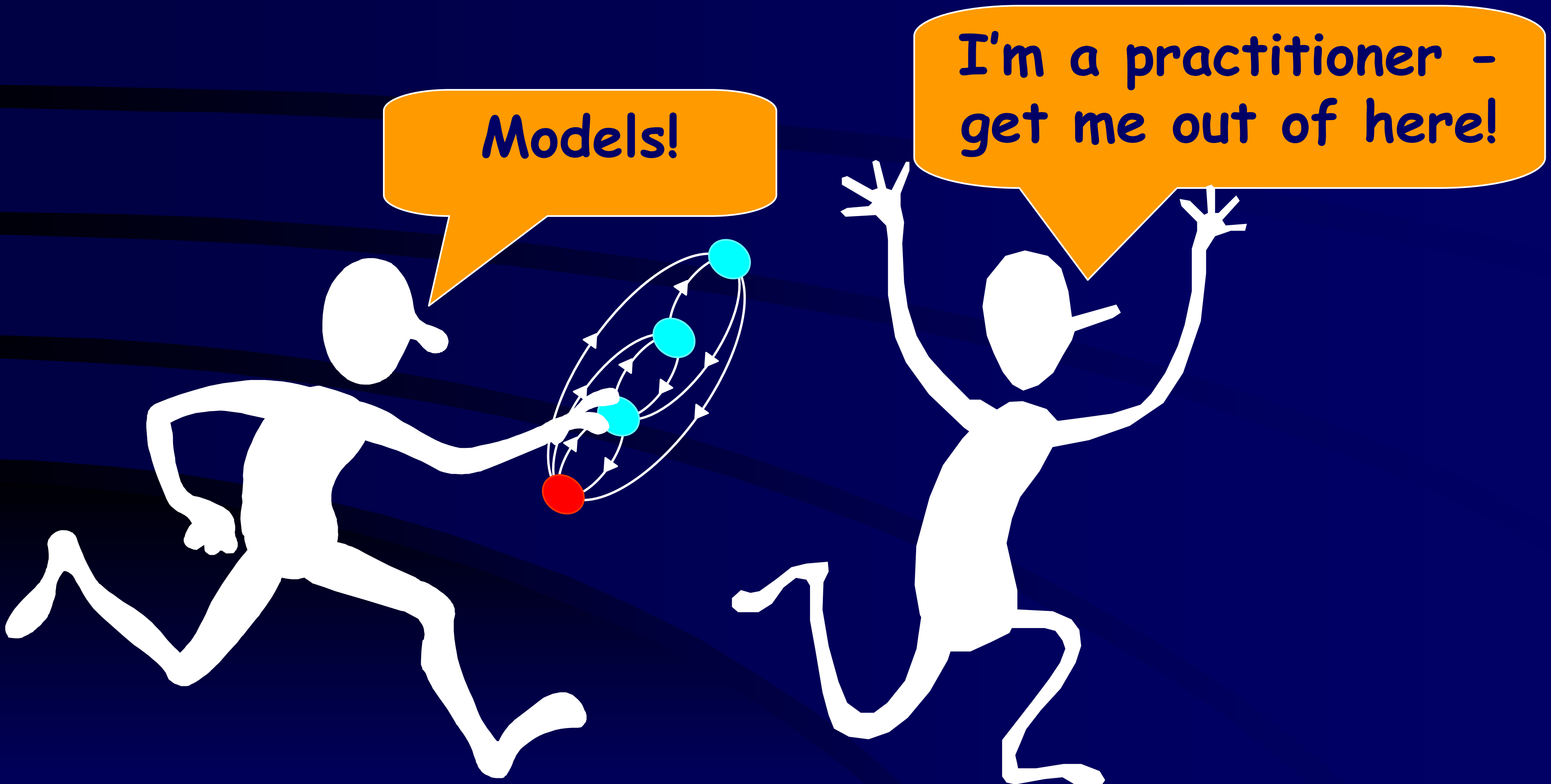
Department of Computing

Imperial College London

Collaborators: R. Chatley, H. Foster, J. Kramer, and J. Magee

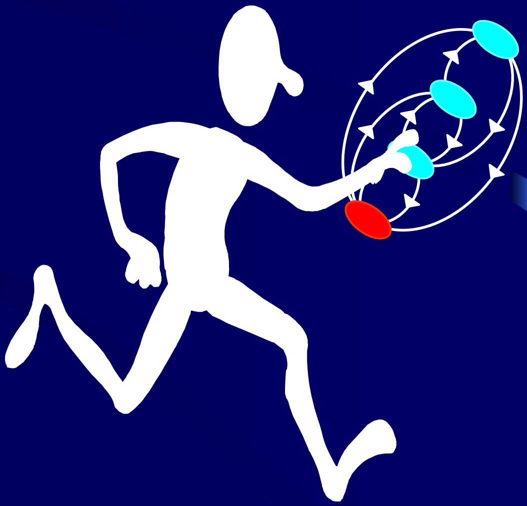
Institut d'Informatique, University of Namur, Feb. 2004

The Problem

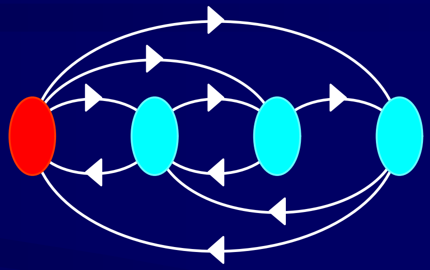


Why models?

- Pre-development analysis of behaviour
  - Prevent consequences
  - Early detection -> cheaper fix
- Traditional engineering approach
  - Abstract & Precise
  - Amenable to analysis.
  - Complexity: Model << System.
- Costs < Benefits



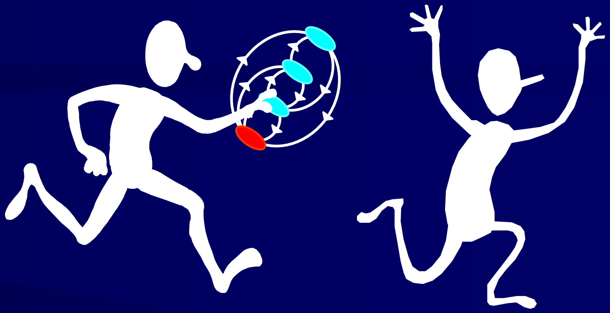
Models for Concurrent & Distributed Systems



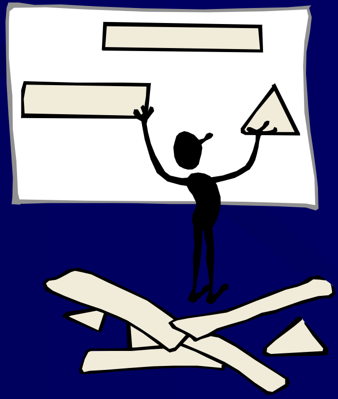
- System structure:
  - Autonomous components.
  - Interactions between them.
- Mathematical foundations
- Amenable to rigorous analysis.
- Effective tool support for analysis
  - model checkers
  - theorem provers
- Successful in uncovering design flaws.

Why NOT models?

- Require expertise
  - Notations
  - Semantics.
- Construction effort is big.
- No benefits until construction is finished.
- Costs > Benefits

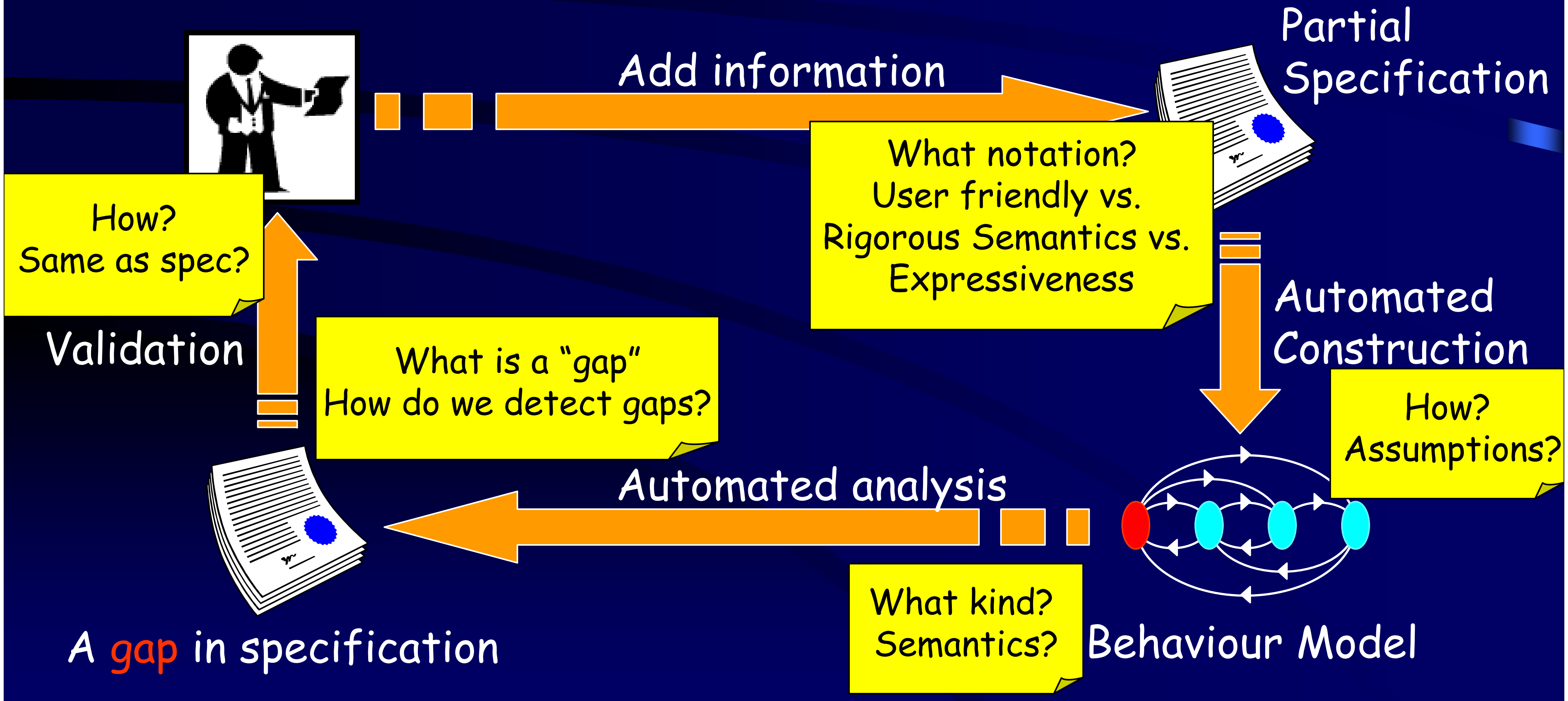


Practitioners prefer informal notations

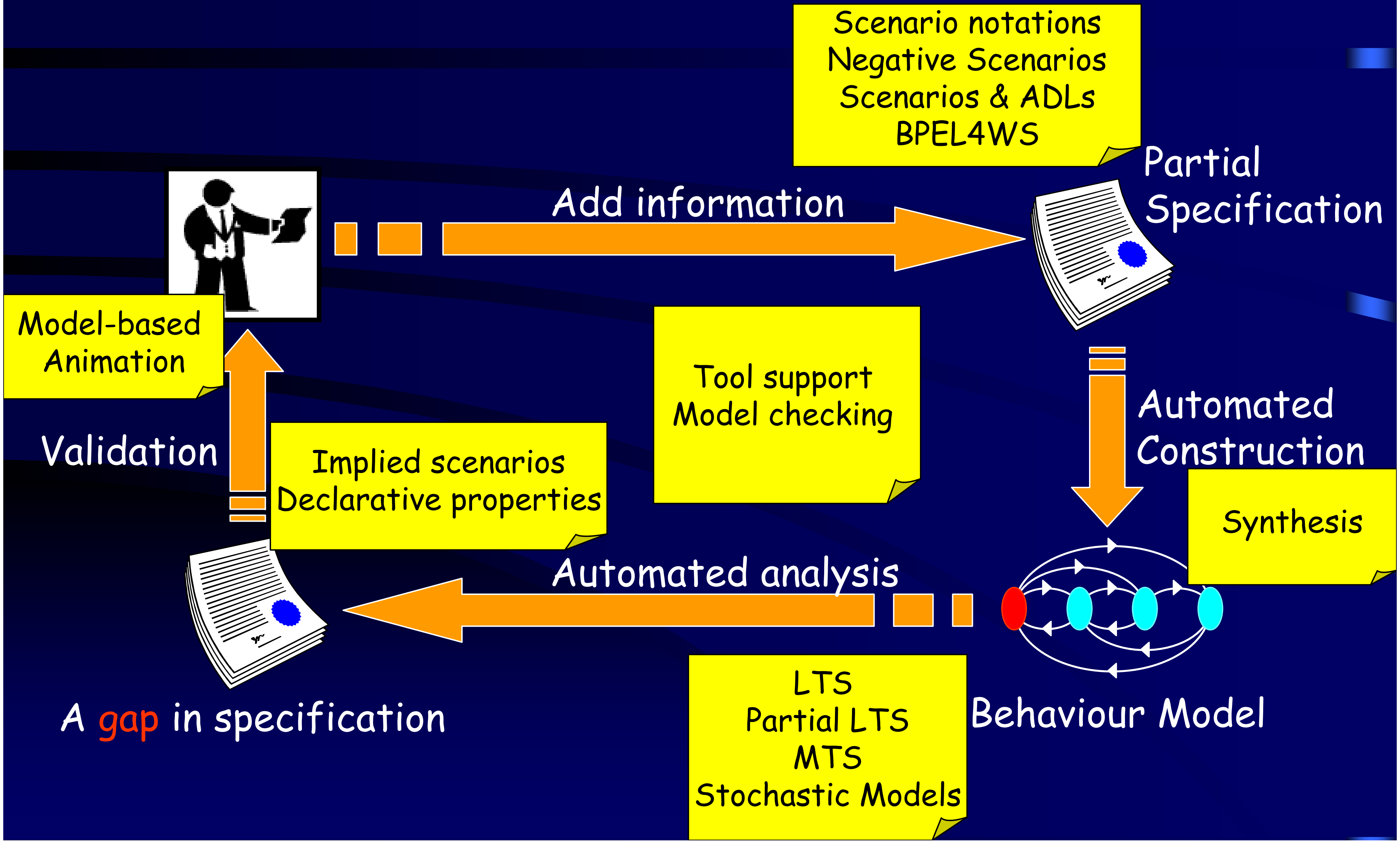


Research Goal

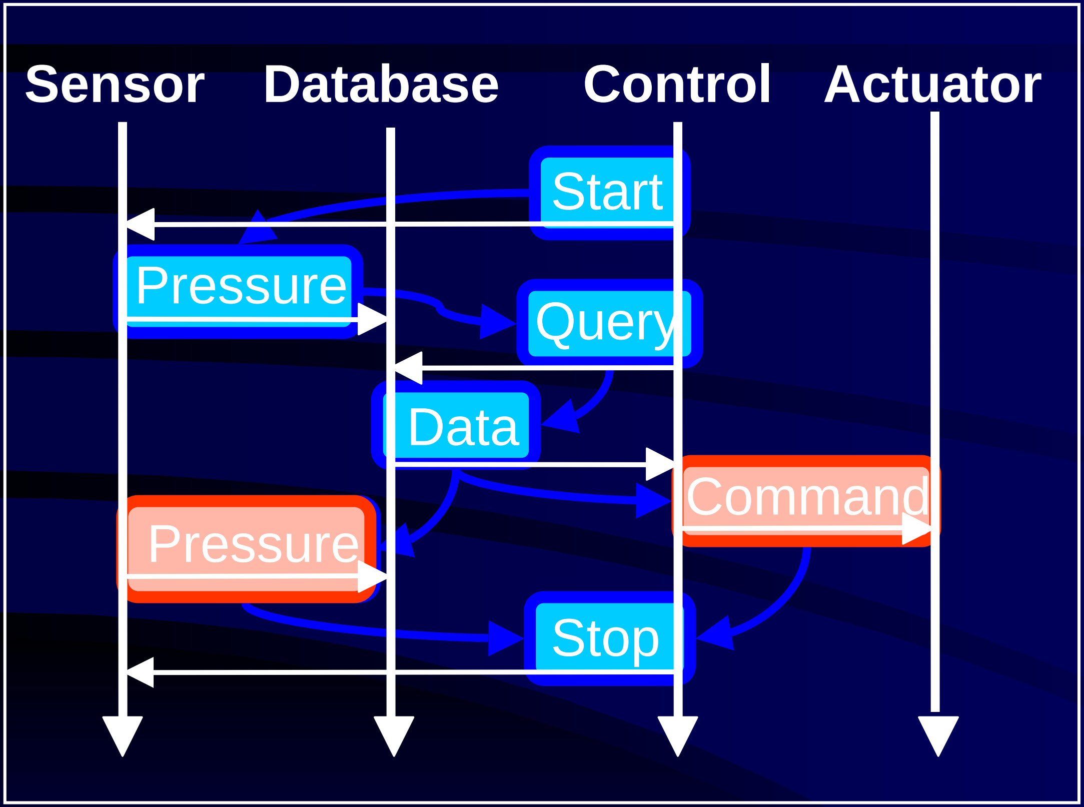
Support the construction and elaboration of behaviour models



Our work: Past, Present and Future



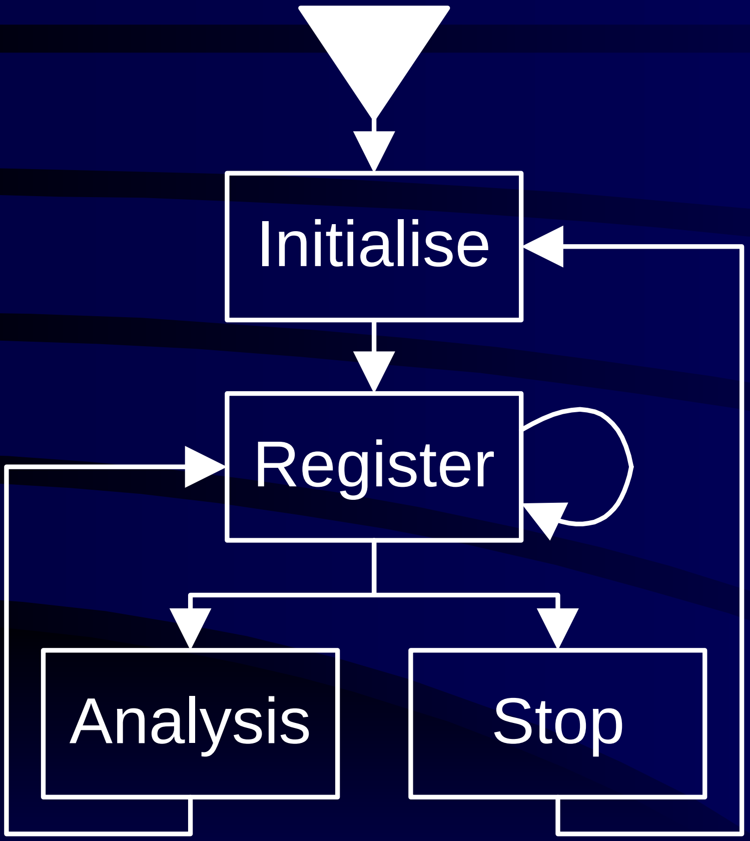
Basic Message Sequence Charts (MSCs)

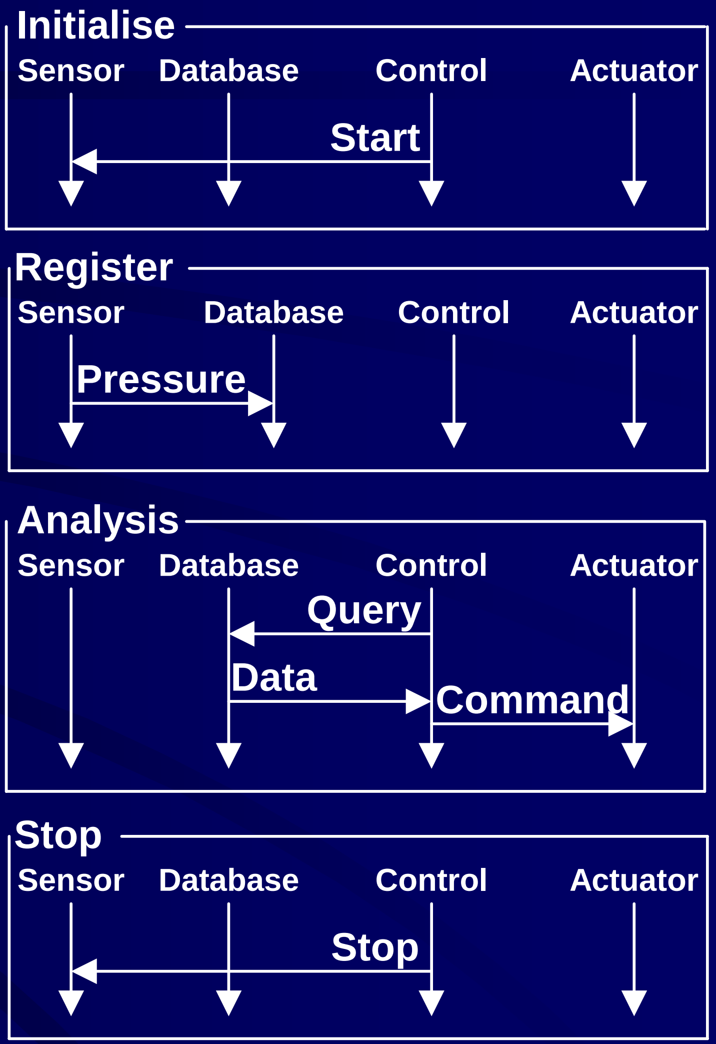


- ITU Z.120 & UML
- Interaction-based
- Partial order semantics.
- Synchronous communication

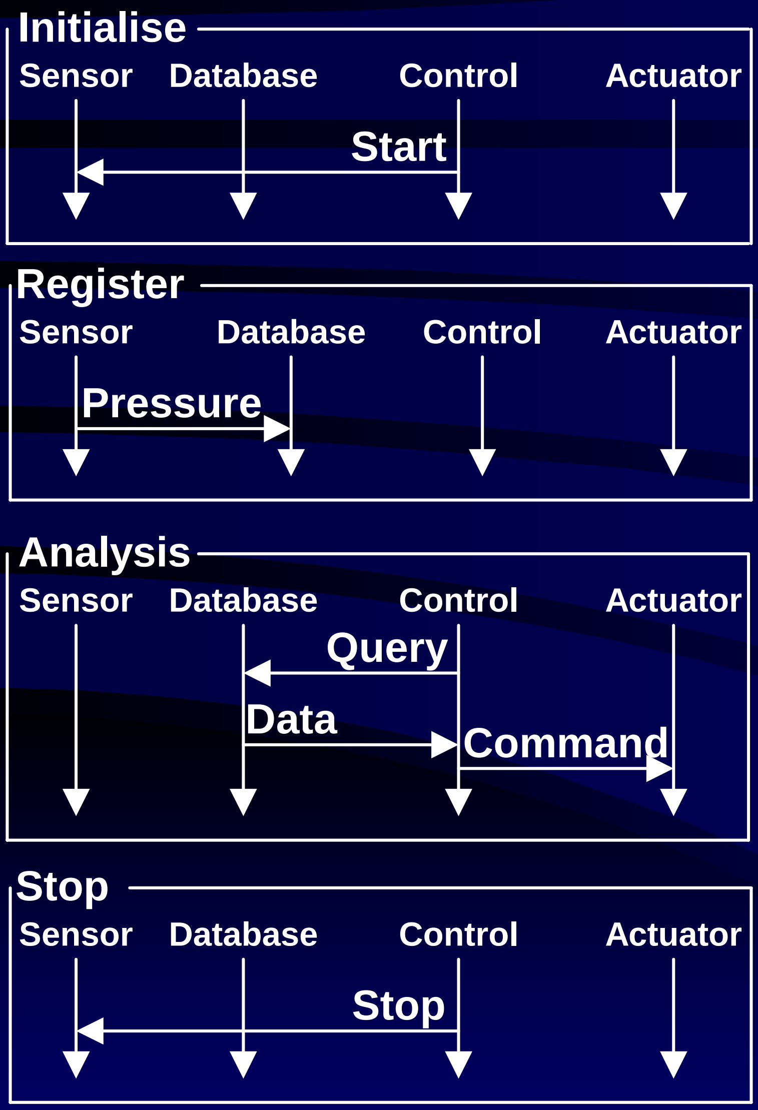
Start, Pressure, Query, Data, **Command**, **Pressure**, Stop.
Start, Pressure, Query, Data, **Pressure**, **Command**, Stop.

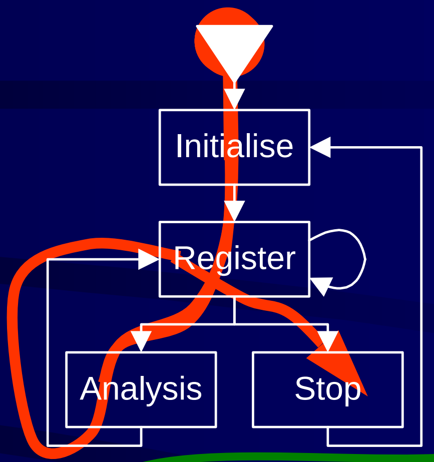
High-level MSCs





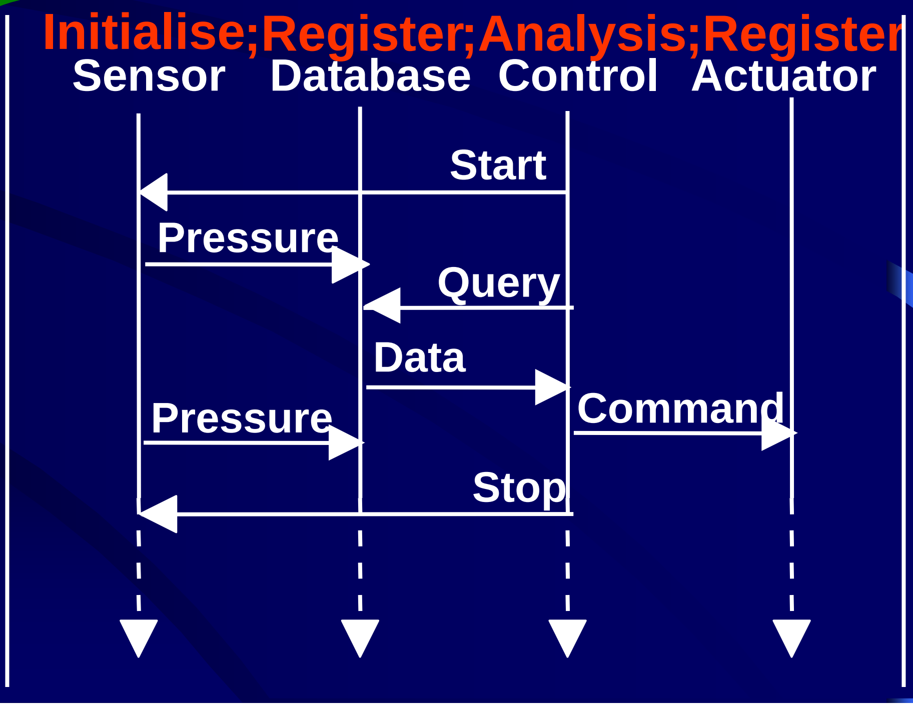
High-level MSC Semantics





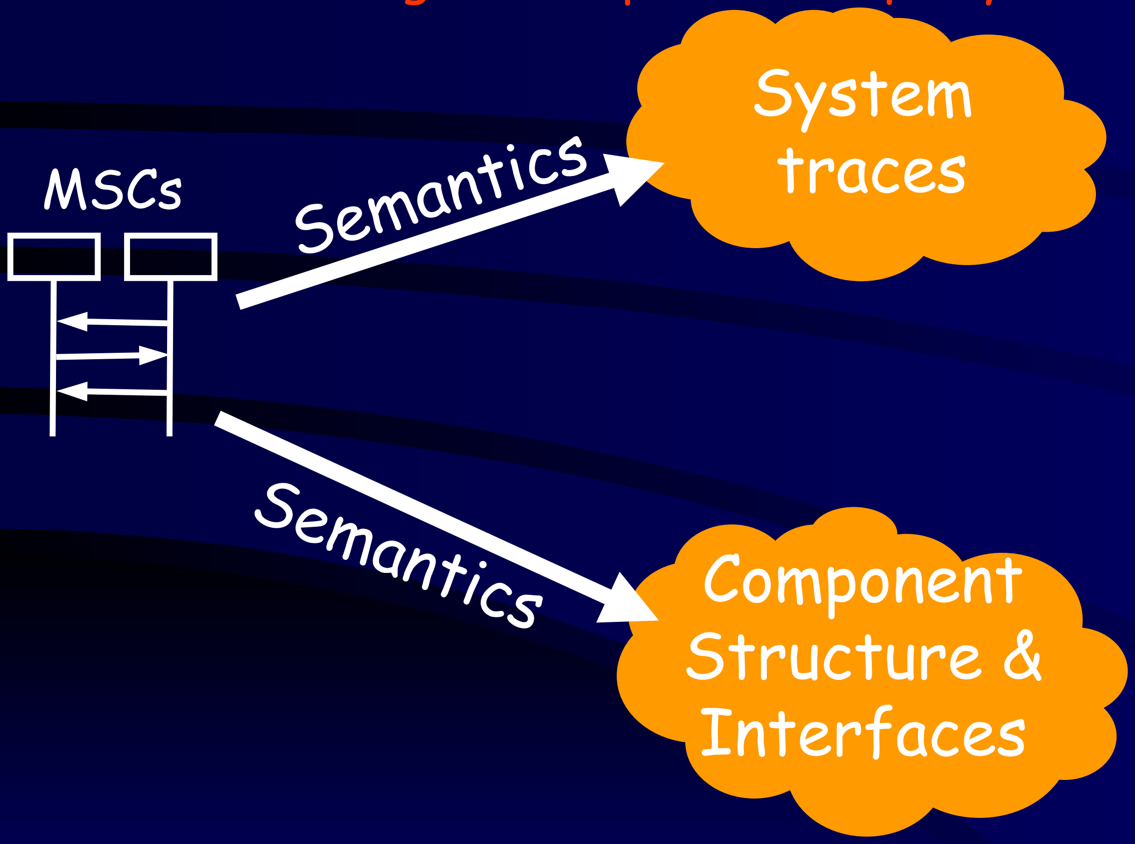
MSC Spec.

Semantics



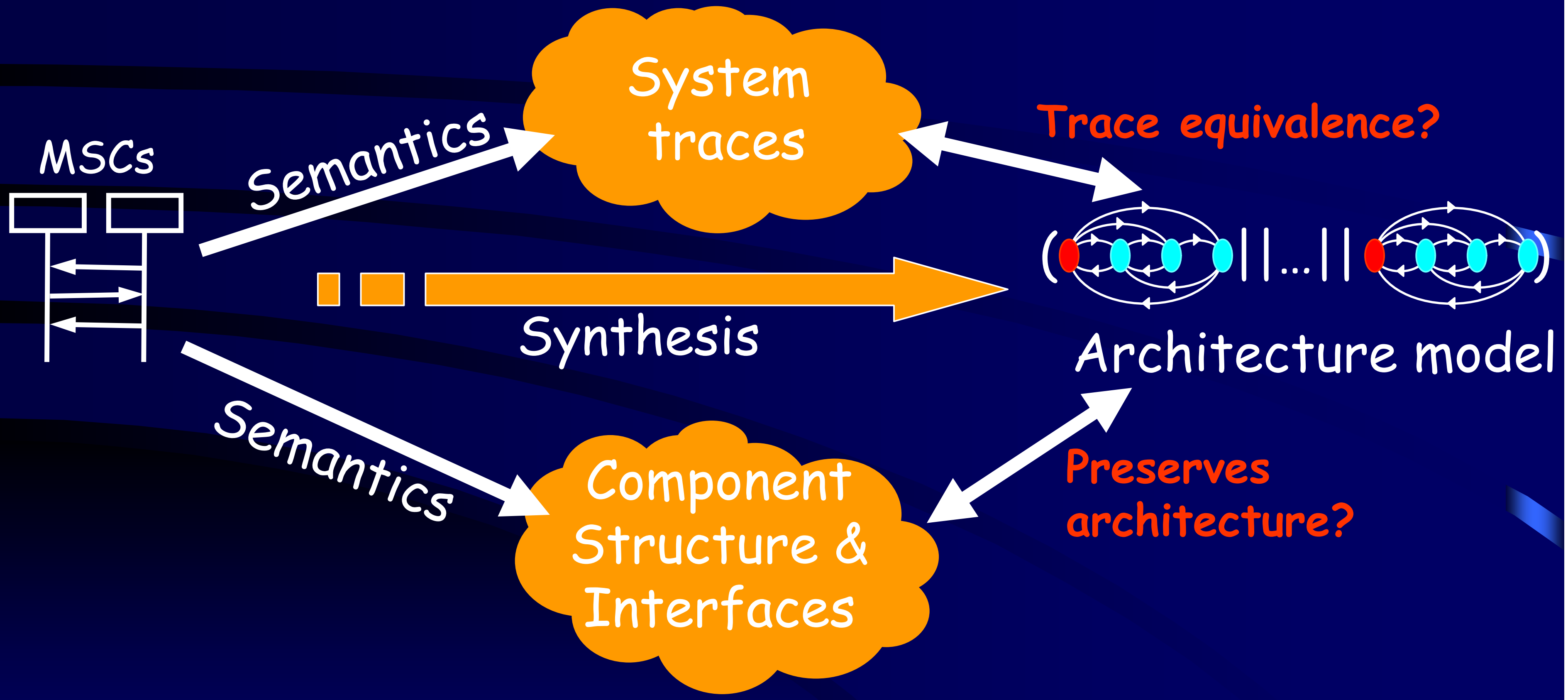
MSC Semantics (Summary)

e.g. start, pressure, query, data, command…

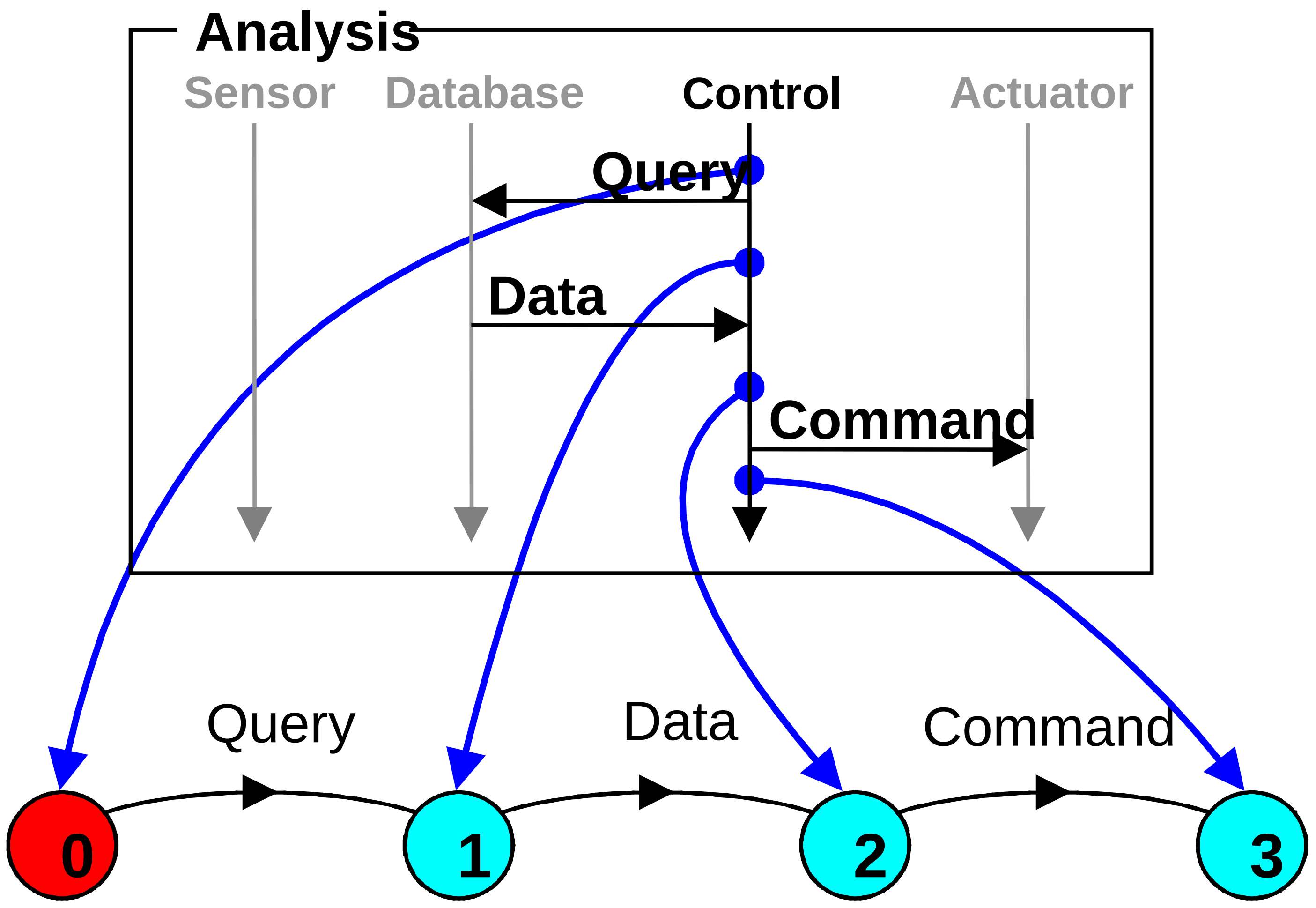


Comps = {Sensor, Database, Control, Actuator}
Database.[query, data, pressure]
Control.[query, data, start, stop]

Model Construction Problem

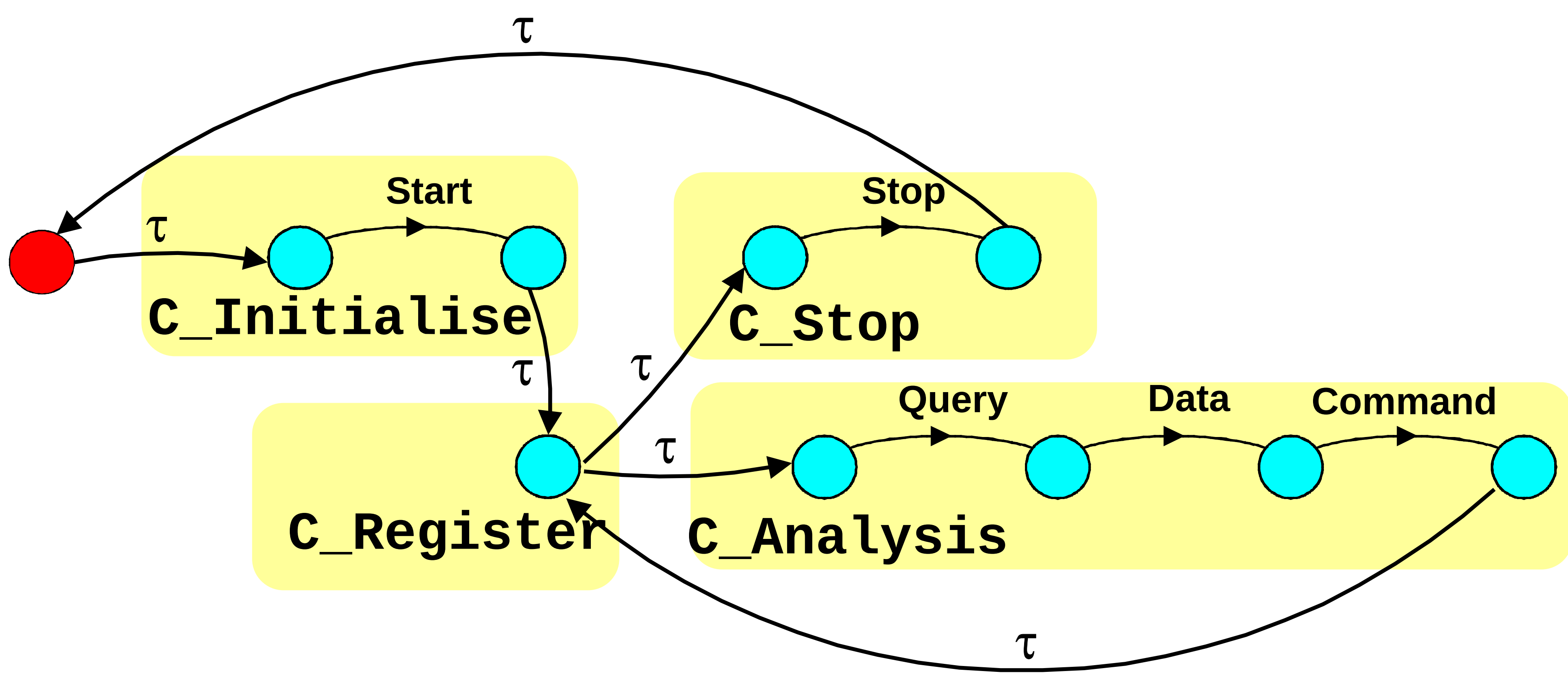


Synthesis of Control Component (1 of 3)



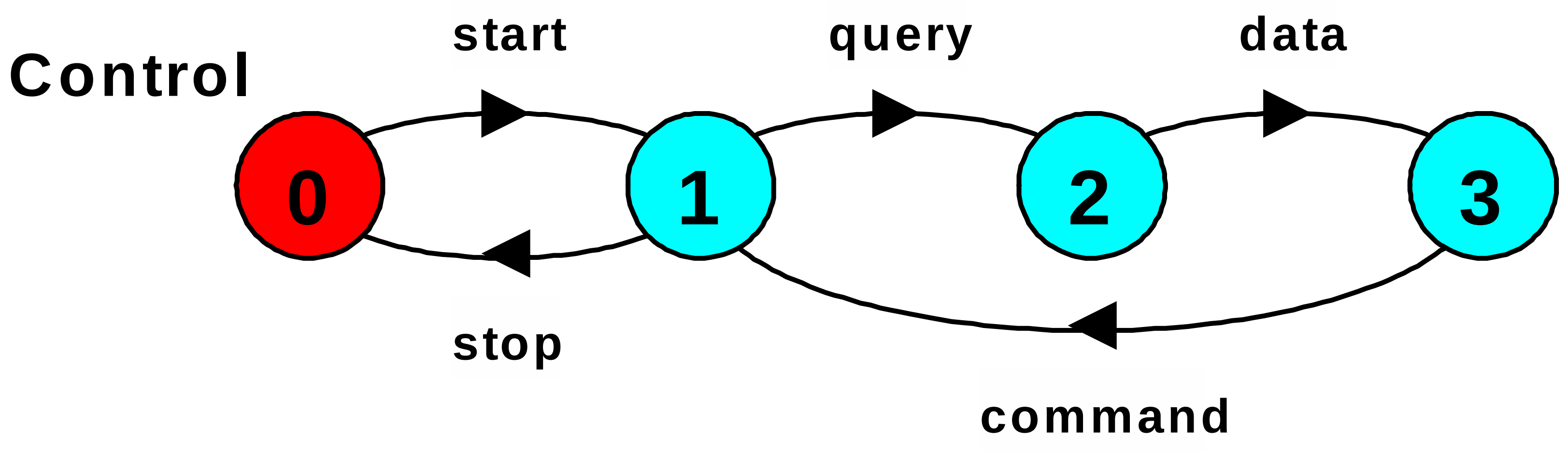
```
C_Analysis = (Query->Data->Command->End)
```

Synthesis of Control Component (2 of 3)



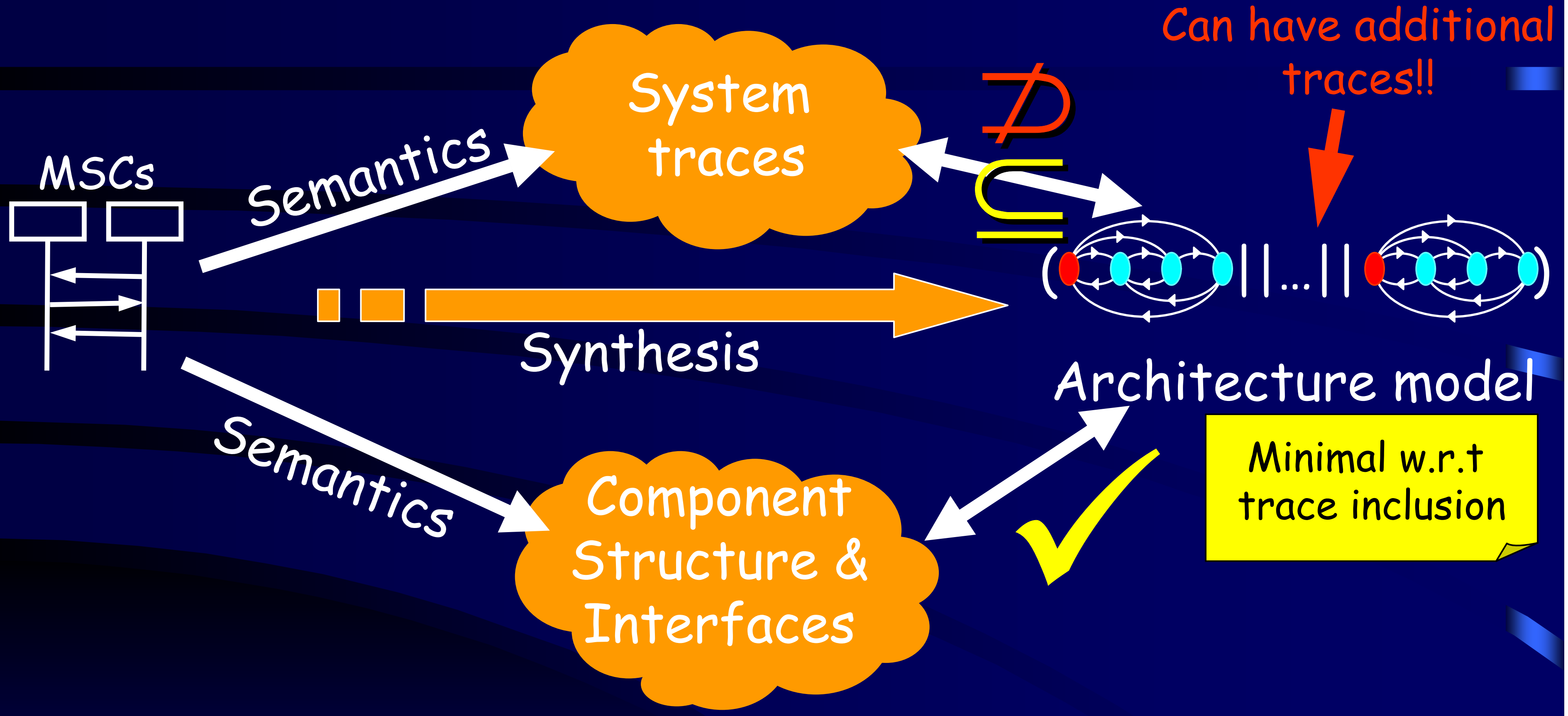
```
Init = C_Initialise,
C_Initialise = C_Register,
C_Register = (t->C_Stop|t->C_Analysis|t->C_Register),
…
```

Synthesis of Control Component(3 of 3)



```
deterministic Control = Init,
Init = Initialise,
Initialise = Register,
Register = (t->Stop|t->Analysis|t->Register),
...
Analysis = (Query->Data->Command->End),
... /{t}
```

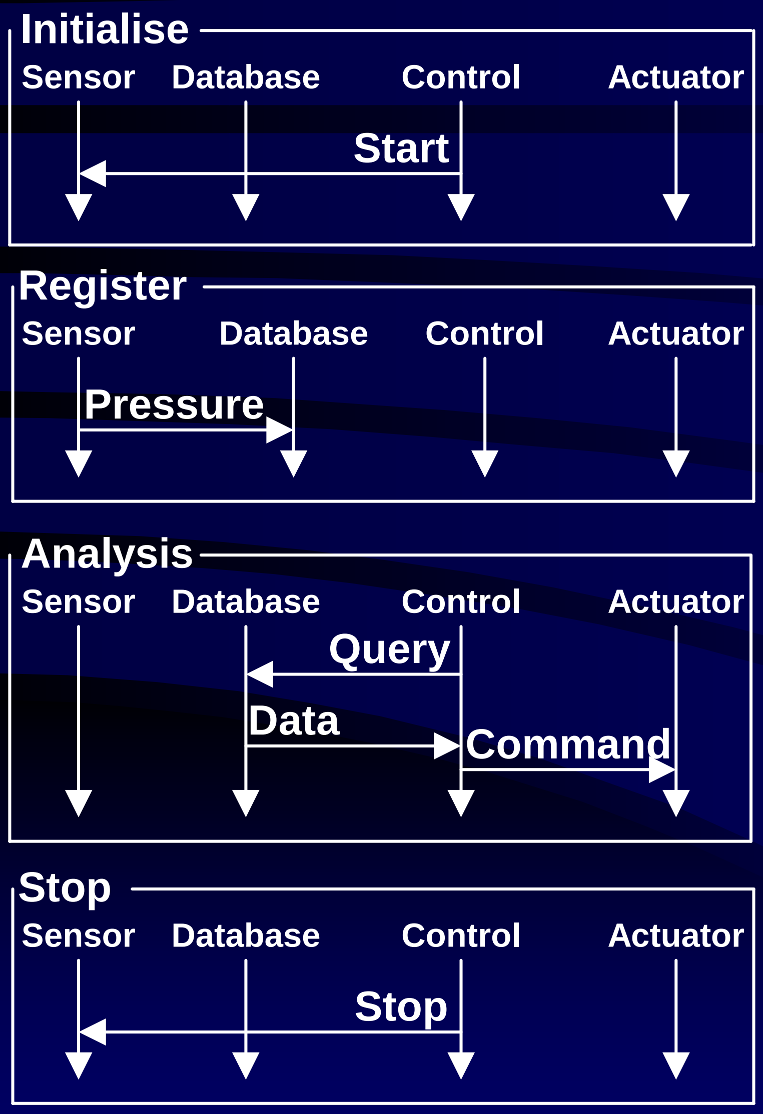
Synthesis Properties

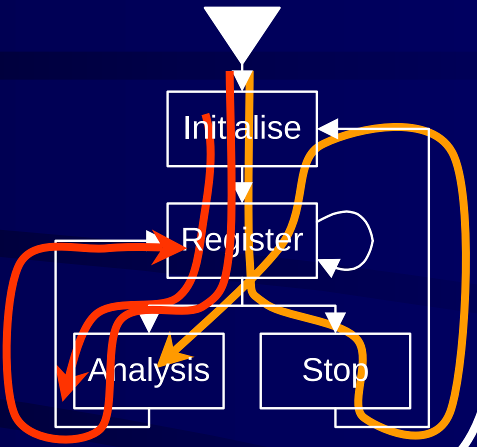

Can have additional traces!!
MSCs
Semantics
System traces
$\not\supset$
$\subseteq$
Synthesis
Architecture model
Semantics
Component Structure & Interfaces
✓
Minimal w.r.t trace inclusion

Implied scenarios
are unspecified traces that appear in all possible architecture models

[FSE'01]

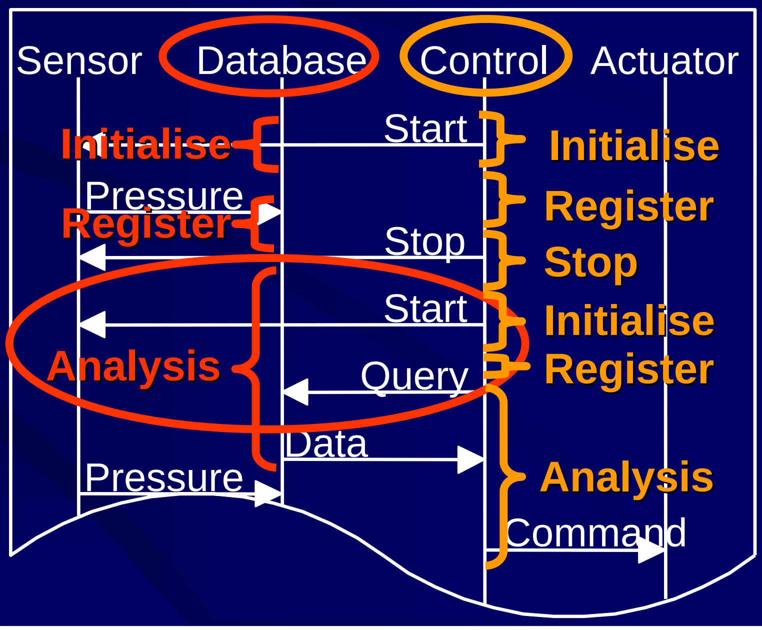
Implied Scenarios: An Example





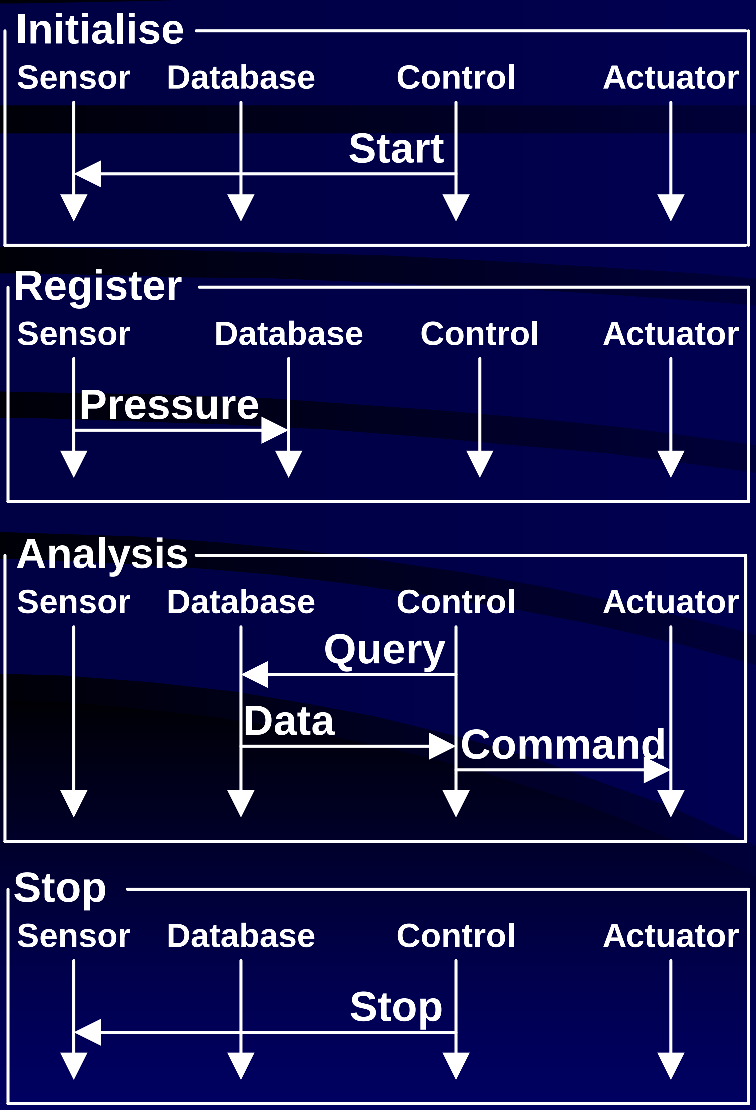
MSC Spec

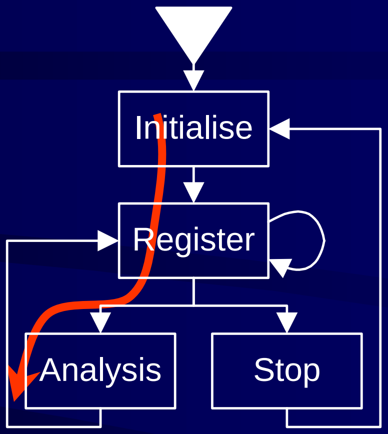
Architecture Model Trace



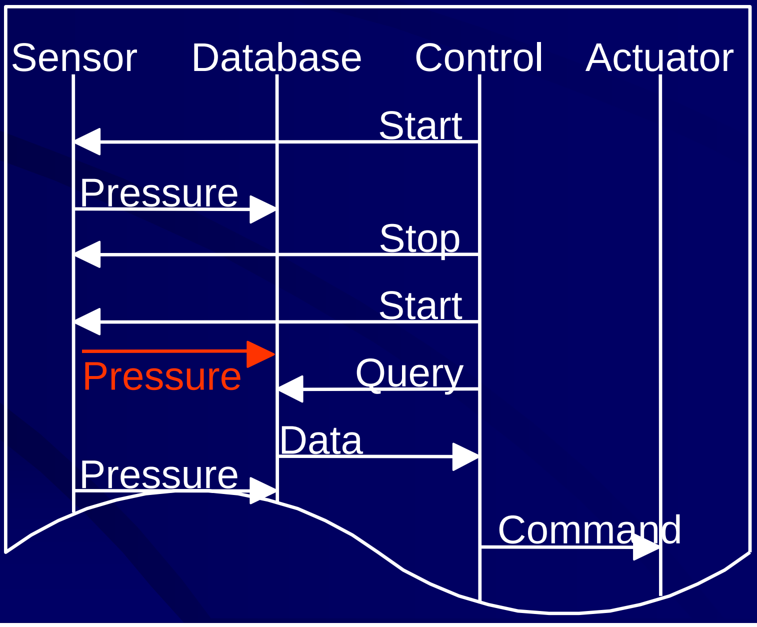
Implied Scenarios: An Example

MSC Spec





Architecture Model Trace



Implied Scenarios…

- Result from a mismatch between specified behaviour and architecture.
- Which one is wrong? Behaviour or Architecture?
  - Missing scenario
  - Incorrect or too abstract architecture
- Implied scenarios are “gaps” in the MSC specification!

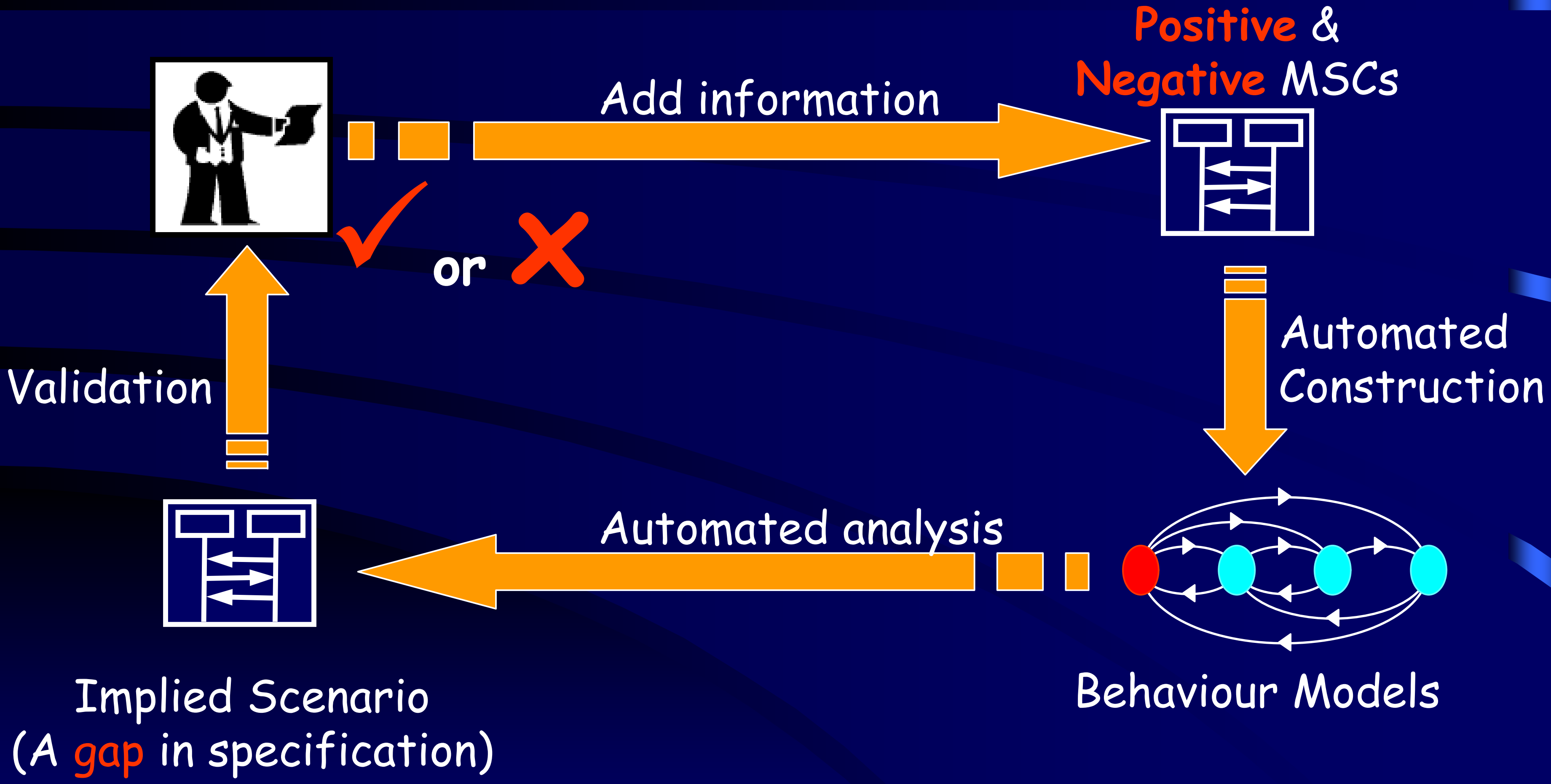
Implied scenarios should be ***detected*** and ***validated***

Implied Scenario Detection

- Build model *Trace Model* T s.t "tr(T)=L(Spec)"
  - Ignore component structure
  - Non-trivial
    - Weak bMSC sequential composition
    - Possibly non-regular MSC language
- Model check "tr(A) $\subseteq$ tr(T)"
  - Declare T as safety property
  - Check for reachability of error state in (T||A)
- Counter-examples are implied scenarios

[FSE'02]

Implied Scenario Validation



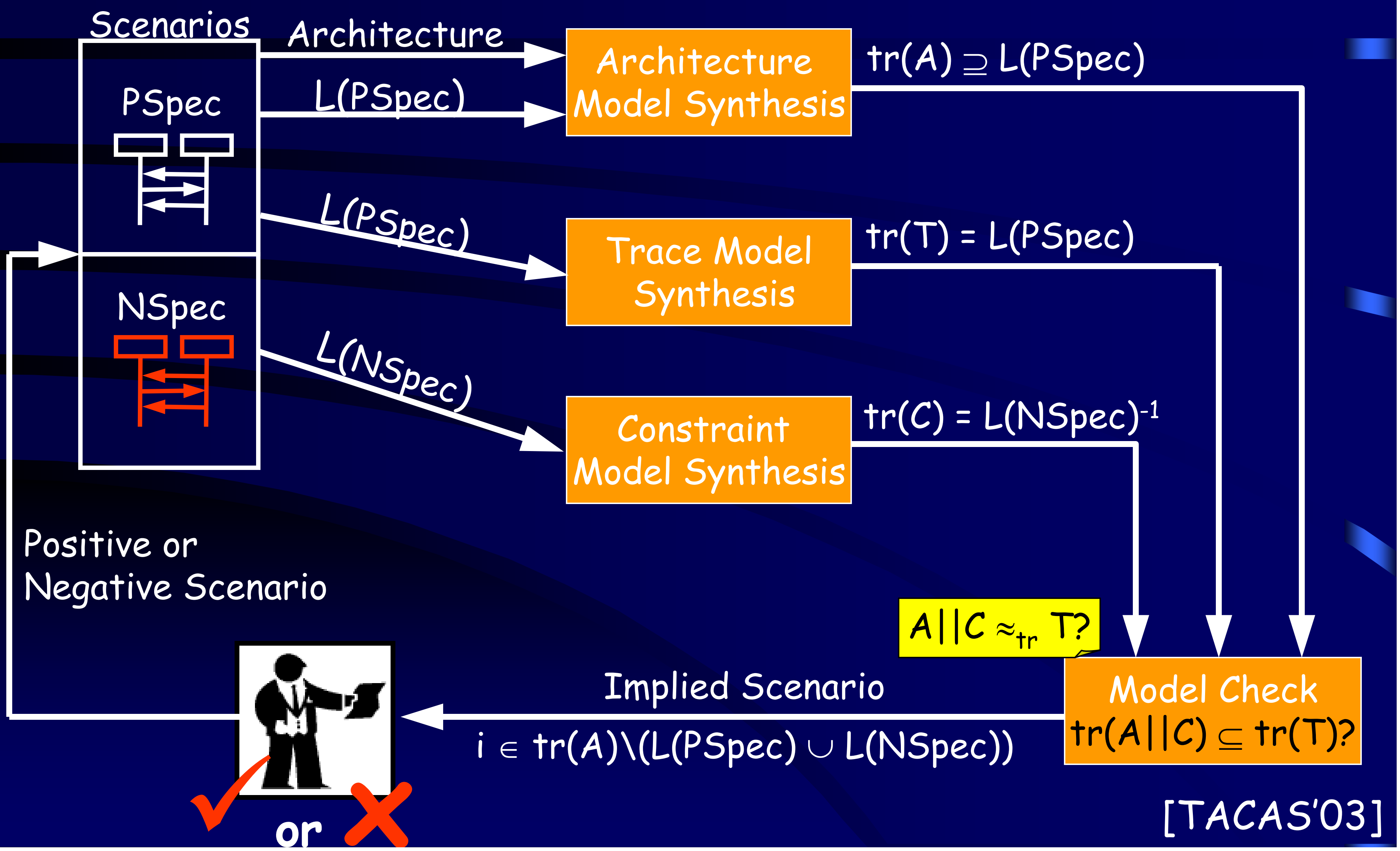
[TOSEM'04]

Negative Scenarios

- Basic Negative Scenarios
  - Allow push-button rejection
  - Reject 1 implied scenario at a time
  - Insufficient to allow process convergence
- Extended Negative Scenarios
  - Abstraction
  - Scope
  - Permit process convergence
  - Require “effort” from user.

[FSE’02]

The Whole Picture

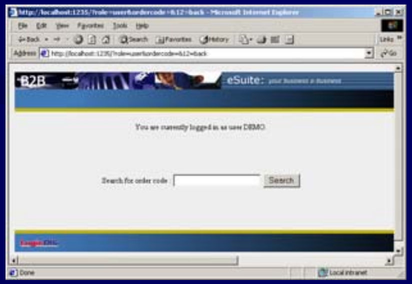


[TACAS'03]

Case Studies

- Railcar Transport System [Harel et al]

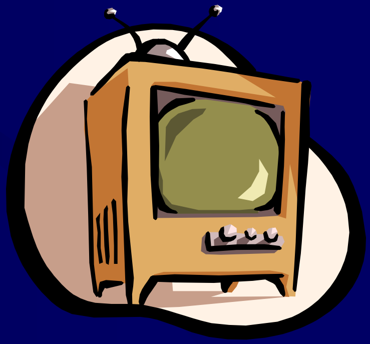
- B2B e-commerce site of greek industrial partners (STATUS project)


http://localhost:1235/?role=user&ordercode=612=back - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost:1235/?role=user&ordercode=612=back
B2B
eSuite: your business in business
Search
Done
Local intranet

You are currently logged in as user DEMO

Search for order code

- Phillips Horizontal Communications Protocol for new product line of television sets.



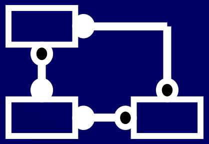
Related Work

- See workshops at OOPSLA'01, ETAPS'01, ICSE'02, ICSE'03, and also Dagstuhl Seminar 03371
- Implied scenarios: Alur, Leue, Protocol synthesis community
- Expressiveness and Model Checking: Peled, Morin,
- Analysis: Muccini, Holzmann, …
- Iterative elaboration: Systa et al.
- Live sequence charts: Harel, Heymans, Bontemps

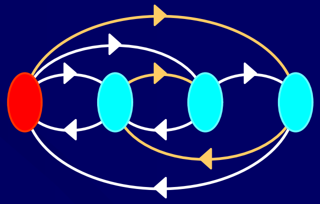
Some Limitations and Open Questions

- Implied scenarios address a very specific aspect of behaviour.
  - Are there other drivers for elaboration?

- Scenarios are instance-level descriptions.
  - Can they be generalised and then used in different settings?



- Synthesis techniques lose the partial nature of scenario specifications.
  - Can we synthesise different kinds of models?



Thank you!

Behaviour Model Construction and Elaboration

Sebastian Uchitel

Department of Computing
Imperial College London