A Notion of Expressiveness for Component-Based Systems

Workshop on Foundations and Applications of Component-based Design

ES Week 2008, Atlanta, 19 October 2008

Joseph Sifakis

in collaboration with Simon Bliudze

VERIMAG Laboratory

Expressiveness for component-based systems

- Comparison between formalisms and models is done by flattening structure and reduction to behaviorally equivalent models e.g. finite state automaton, Turing machine

- This leads to notions of expressiveness that are **not adequate** for comparing coordination capabilities of languages and models e.g.
  - ❑ all finite state formalisms turn out to be expressively equivalent
  - ❑ all modeling and programming languages are Turing complete, while their coordination capabilities tremendously differ

Objectives:

- Propose notions of expressiveness based on a strict separation between behavior and coordination
- Compare existing frameworks by using such notions

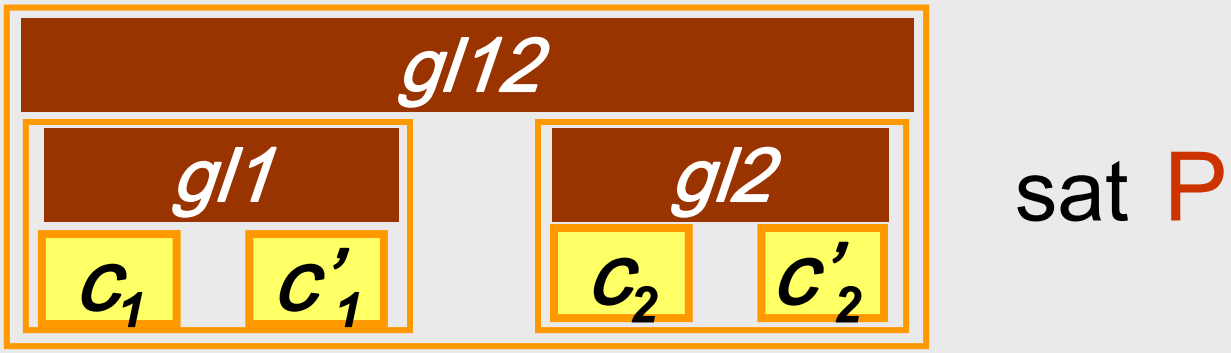
OVERVIEW

- Component-based Construction
- BIP: Basic Concepts
- Expressiveness
- Methodology for Componentization
- Discussion

Component-based Construction: The Problem

Build a component *C* satisfying a given property *P, from*

- $\mathcal{C}_0$ a set of **atomic** components described by their behavior
- $\mathcal{GL} = \{gl_1, \ldots, gl_i, \ldots\}$ a set of **glue *operators*** on components

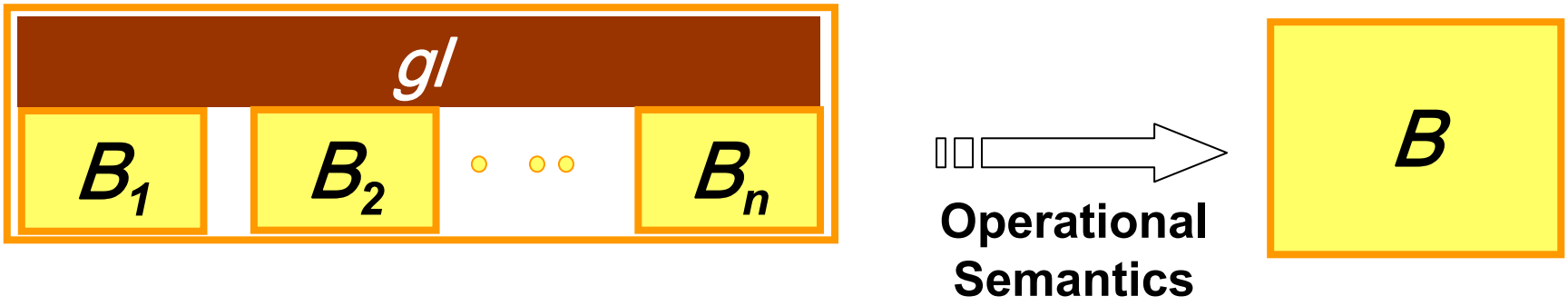


- Move from frameworks based on single composition operators to frameworks based on families of composition operators

- Glue operators allow modeling coordination mechanisms such as such as protocols, schedulers, buses

Glue operators: operational semantics

We use operational semantics to define the meaning of a composite component – glue operators are “behavior transformers”



Glue Operators

- build interactions of composite components from the actions of the atomic components e.g. parallel composition operators
- can be specified by using a family of derivation rules (the Universal Glue)

Glue operators

A **glue operator** is a set of derivation rules of the form

$$\frac{\{q_i - a_i \rightarrow q'_i\}_{i \in I} \quad \{\neg\, q_k - a_k \rightarrow\}_{k \in K}}{(q_1, .., q_n) - a \rightarrow (q'_1, .., q'_n)}$$

- $I,K \subseteq \{1, \ldots n\}$, $I \neq \emptyset$, $K \cap I = \emptyset$
- $a = \bigcup_{i \in I} a_I$ is an interaction
- $q'_i = q_i$ for $i \notin I$

Notice that, non deterministic choice and sequential composition are not glue operators

A **glue** is a set of glue operators

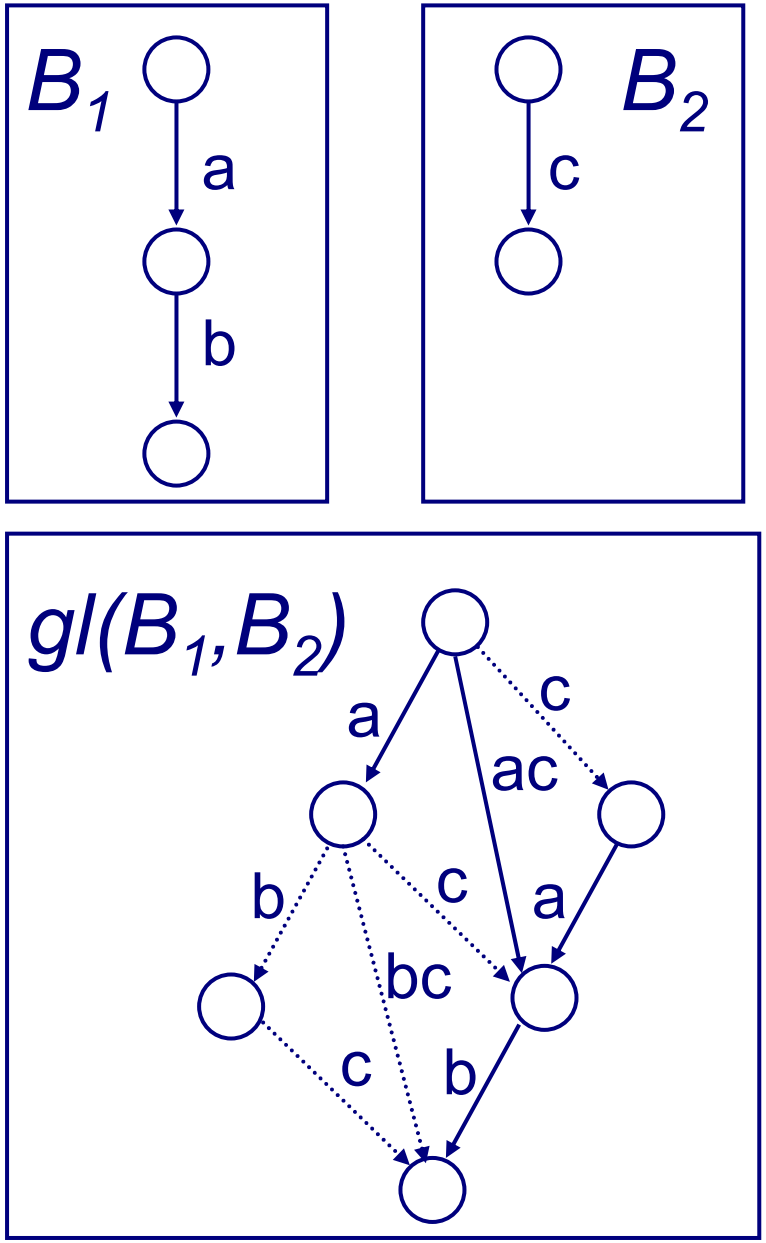
Glue operators: Example

gl is defined by

$$\frac{q_1 - a \rightarrow q'_1}{q_1 q_2 - a \rightarrow q'_1 q_2}$$

$$\frac{q_1 - a \rightarrow q'_1 \quad q_2 - c \rightarrow q'_2}{q_1 q_2 - ac \rightarrow q'_1 q'_2}$$

$$\frac{q_1 - b \rightarrow q'_1 \quad \neg\, q_2 - c \rightarrow}{q_1 q_2 - b \rightarrow q'_1 q_2}$$

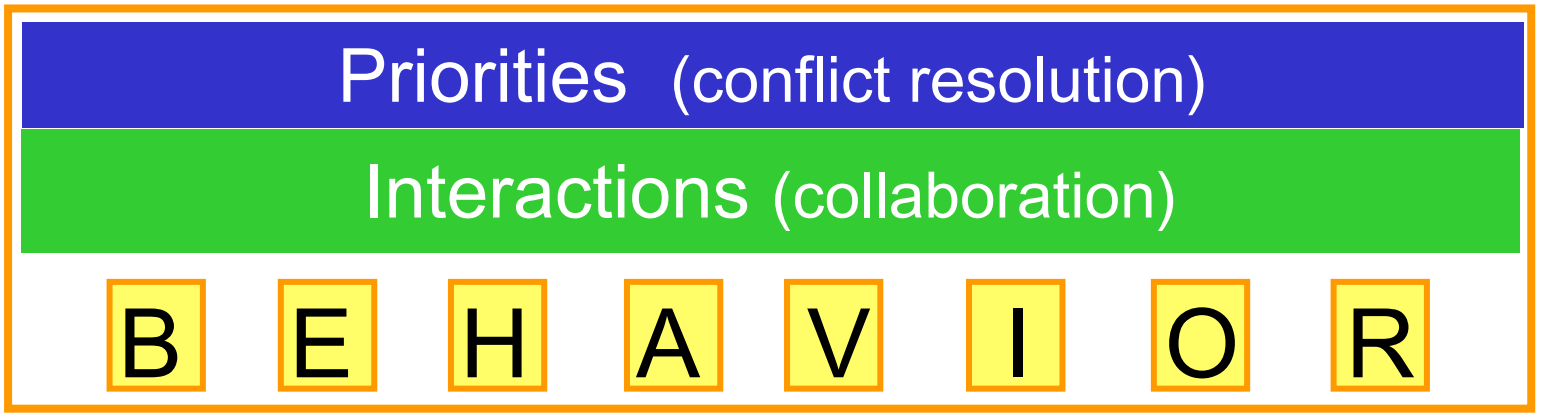


OVERVIEW

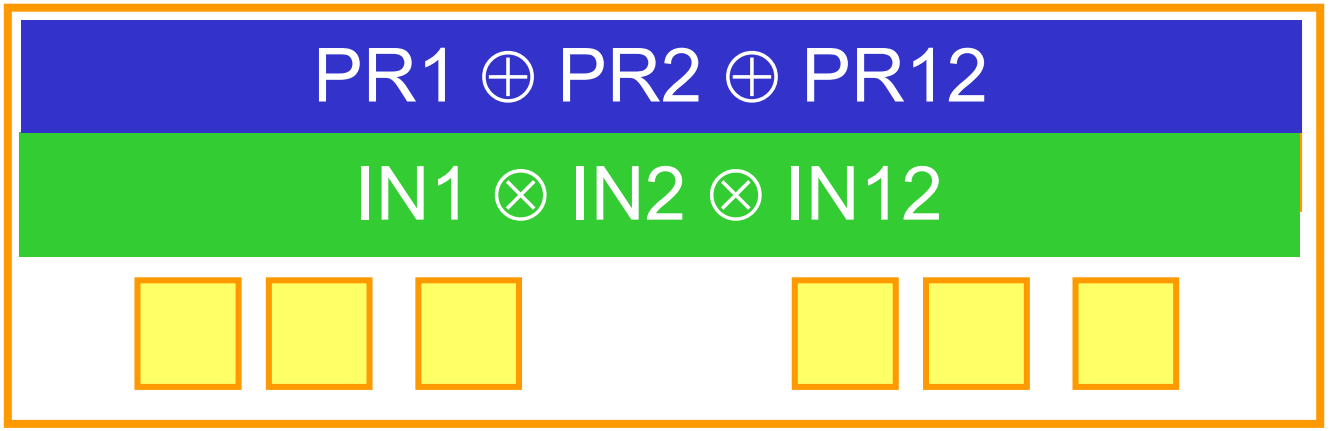
- Component-based Construction
- BIP: Basic Concepts
- Expressiveness
- Methodology for Componentization
- Discussion

BIP: Basic Concepts

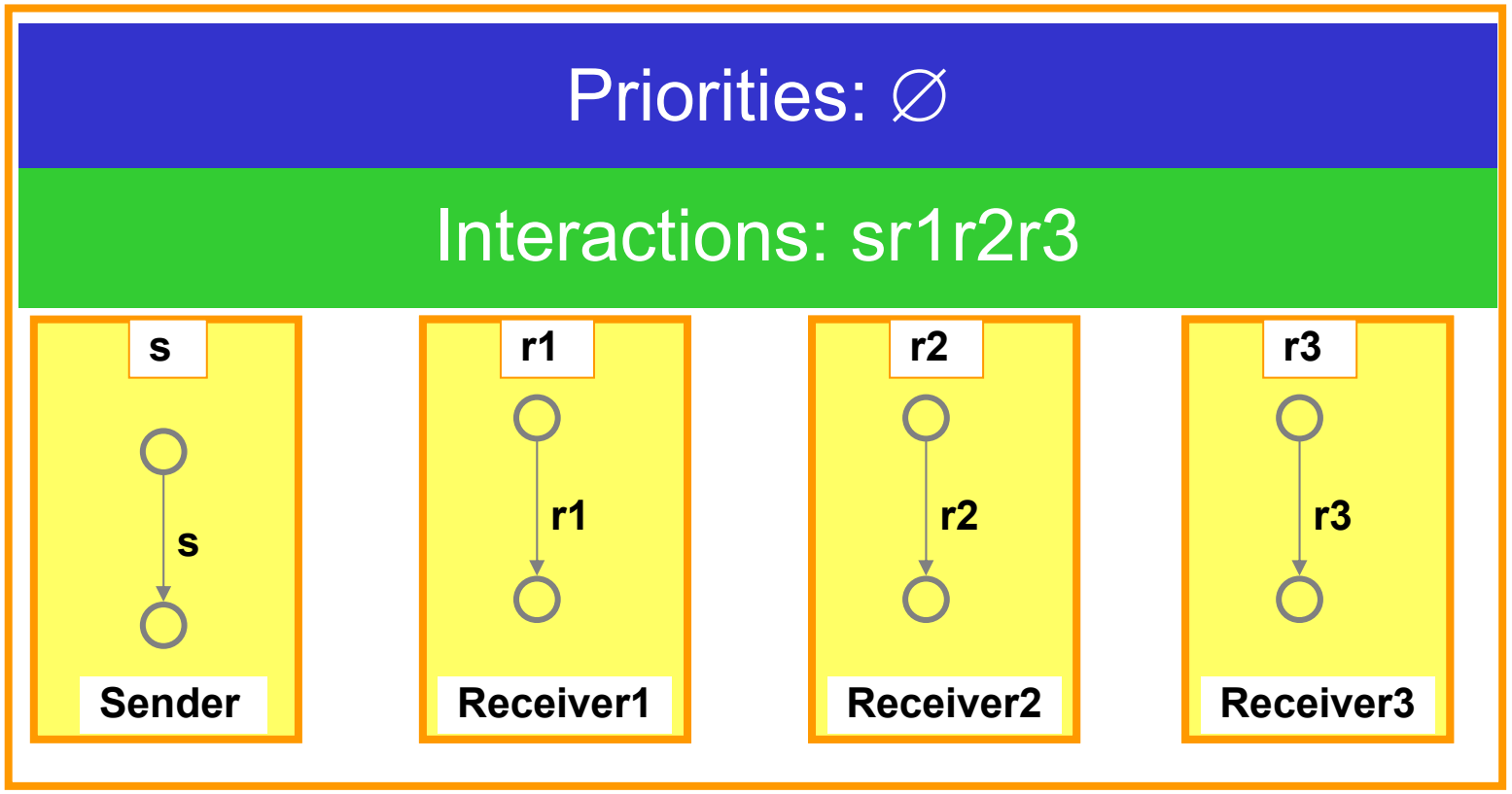
Layered component model



Composition operation parameterized by glue IN12, PR12

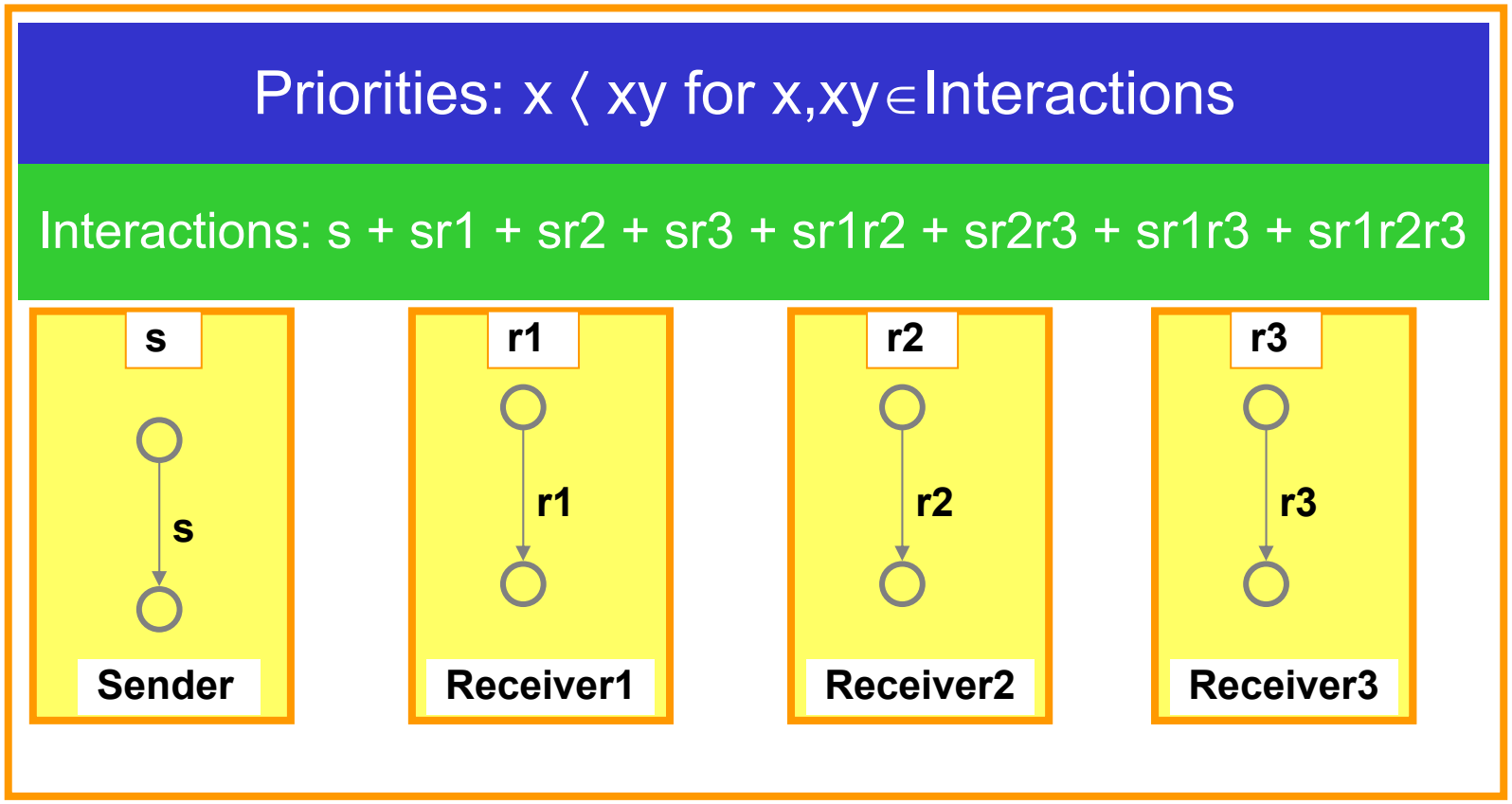


BIP: Basic Concepts



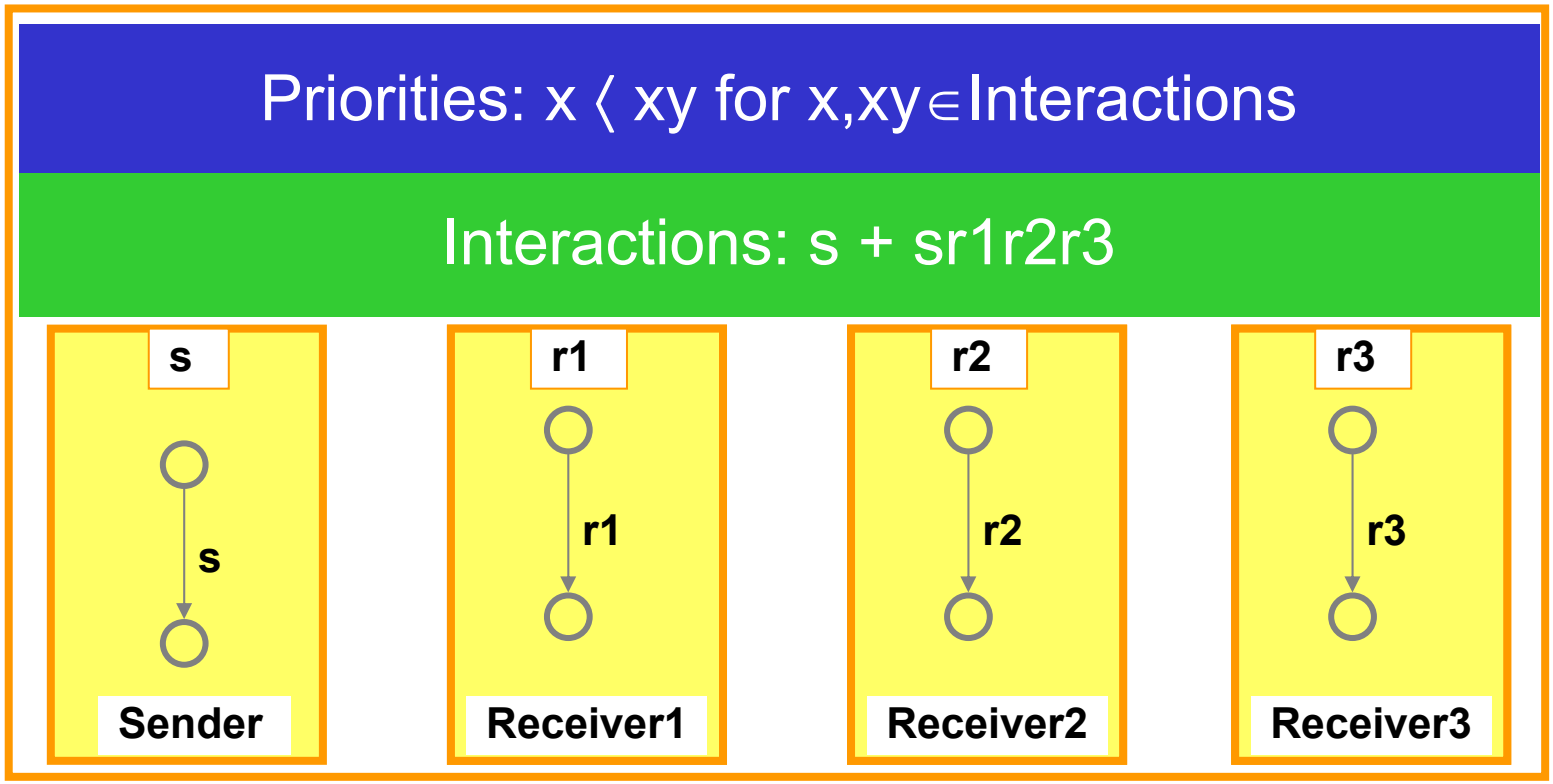
Rendezvous

BIP: Basic Concepts



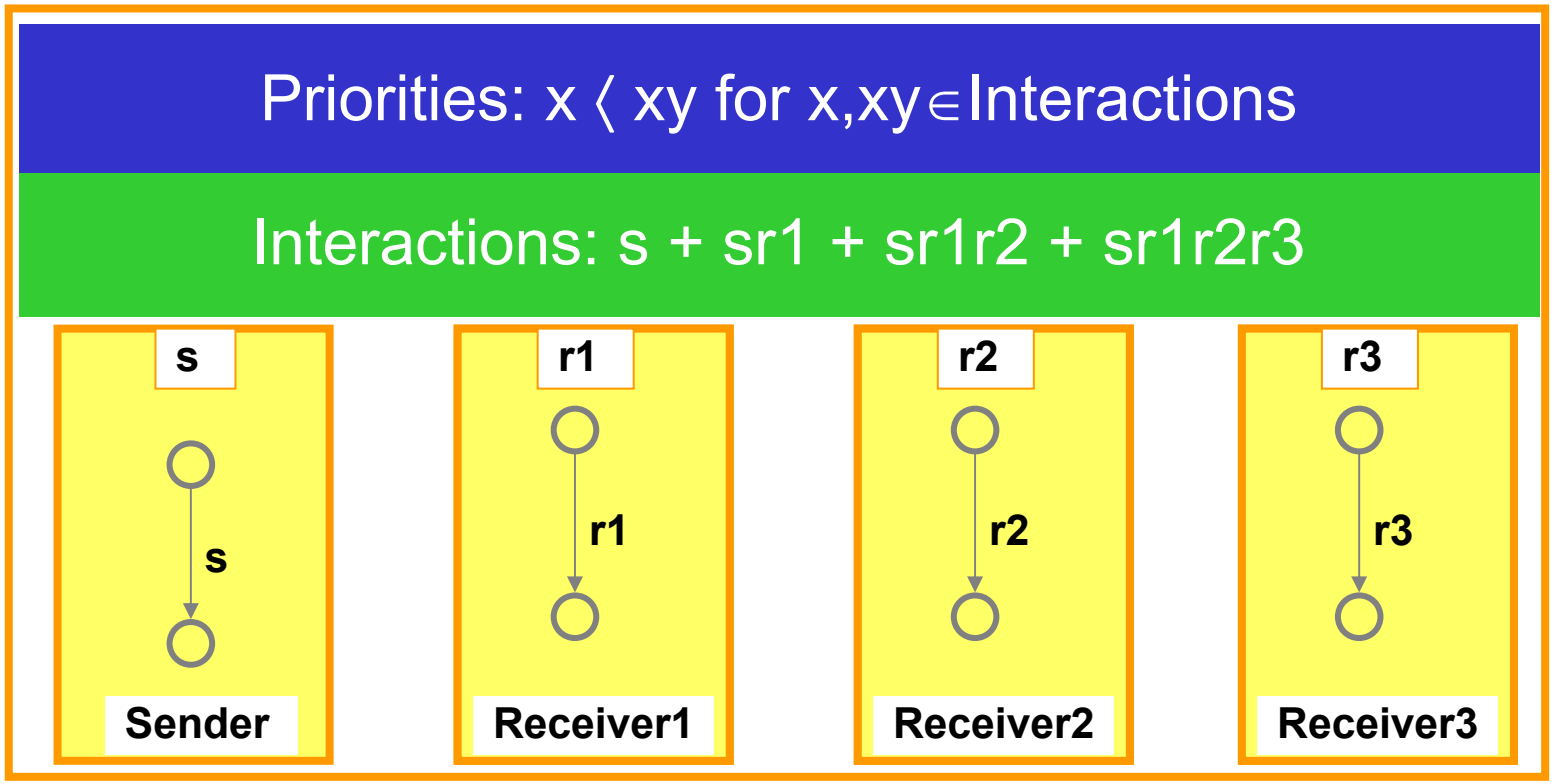
Broadcast

BIP: Basic Concepts



Atomic Broadcast

BIP: Basic Concepts



Causal Chain

BIP: Basic Concepts: Semantics

- a set of atomic components $\{B_i\}_{i=1..n}$ where $B_i = (Q_i, 2^{Pi}, \rightarrow_i)$
- a set of interactions $\gamma \in 2^{2P}$ with $P = \cup_{i=1..n} P_i$ and $P_i \cap P_j = \emptyset$ $P = \cup_{i=1..n} P_i$
- a strict partial order $\pi \subseteq 2^P \times 2^P$

$\}$ $\pi\, \gamma\, (B1,.,\, Bn)$

Interactions

$$\frac{\{a_i\}_{i\in I} \in \gamma \quad \{q_i - a_i \rightarrow q'_i\}_{i\in I} \quad a = \cup_{i\in I} a_i}{(q_1,.,\, q_n) - a \rightarrow_\gamma (q'_1,.,\, q'_n) \text{ where } q'_I = q_I}$$

Priorities

$$\frac{q - a \rightarrow_\gamma q' \quad \neg\, (\exists\, q - b \rightarrow_\gamma \wedge\, a\, \pi\, b\,)}{q - a \rightarrow_\pi q'}$$

OVERVIEW

- Component-based Construction
- BIP: Basic Concepts
- Expressiveness
- Methodology for Componentization
- Discussion

Expressiveness for Component-based Systems

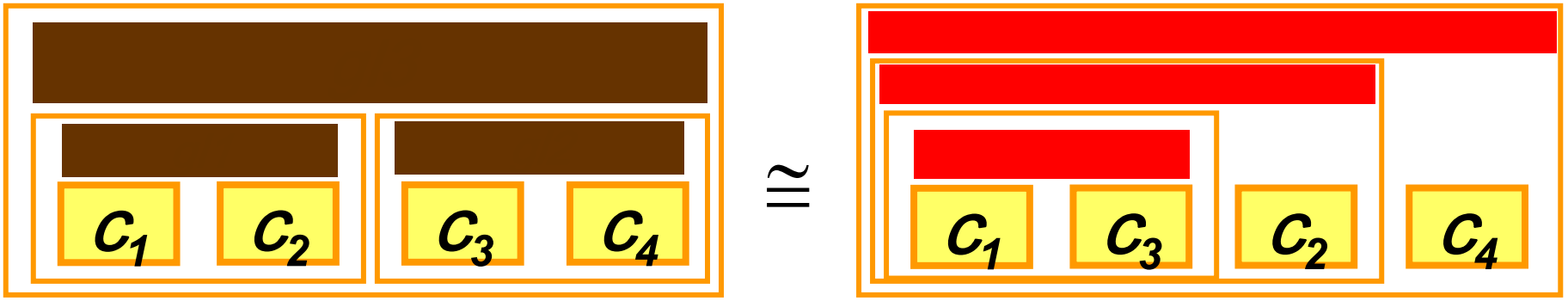
- Different from the usual notion of expressiveness!
- Based on strict separation between glue and behavior

Given two glues G_1 , G_2

G_2 ***is strongly more expressive than*** G_1

if for any component built by using G_1 and $\mathcal{C}_0$

there exists an equivalent component built by using G_2 and $\mathcal{C}_0$



Expressiveness for component-based systems

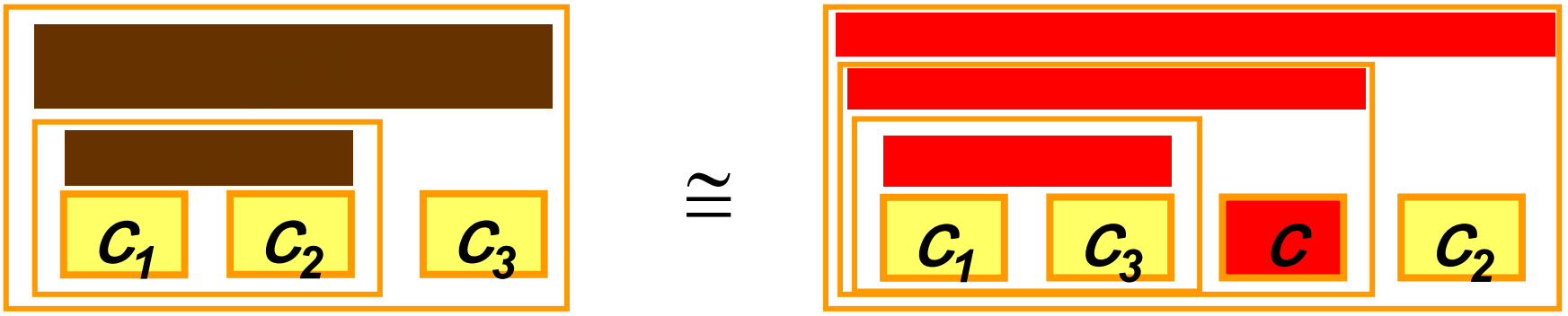
Given two glues G_1 , G_2

G_2 ***is weakly more expressive than*** G_1

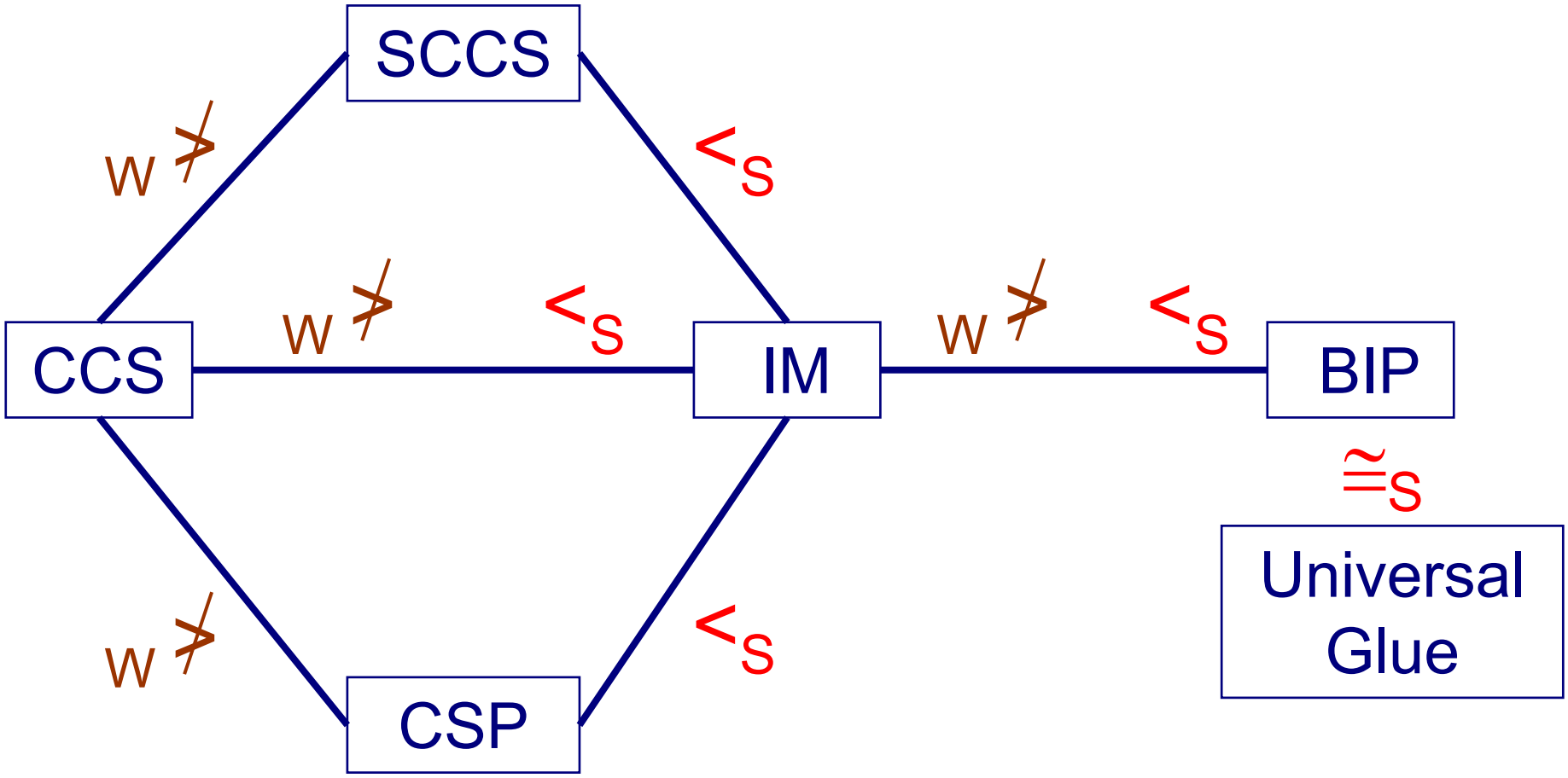
if for any component built by using G_1 and $\mathscr{C}_0$

there exists an equivalent component built by using G_2 and $\mathscr{C}_0 \cup \mathscr{C}$

where $\mathscr{C}$ is a finite set of coordination behaviors.



Expressiveness for component-based systems

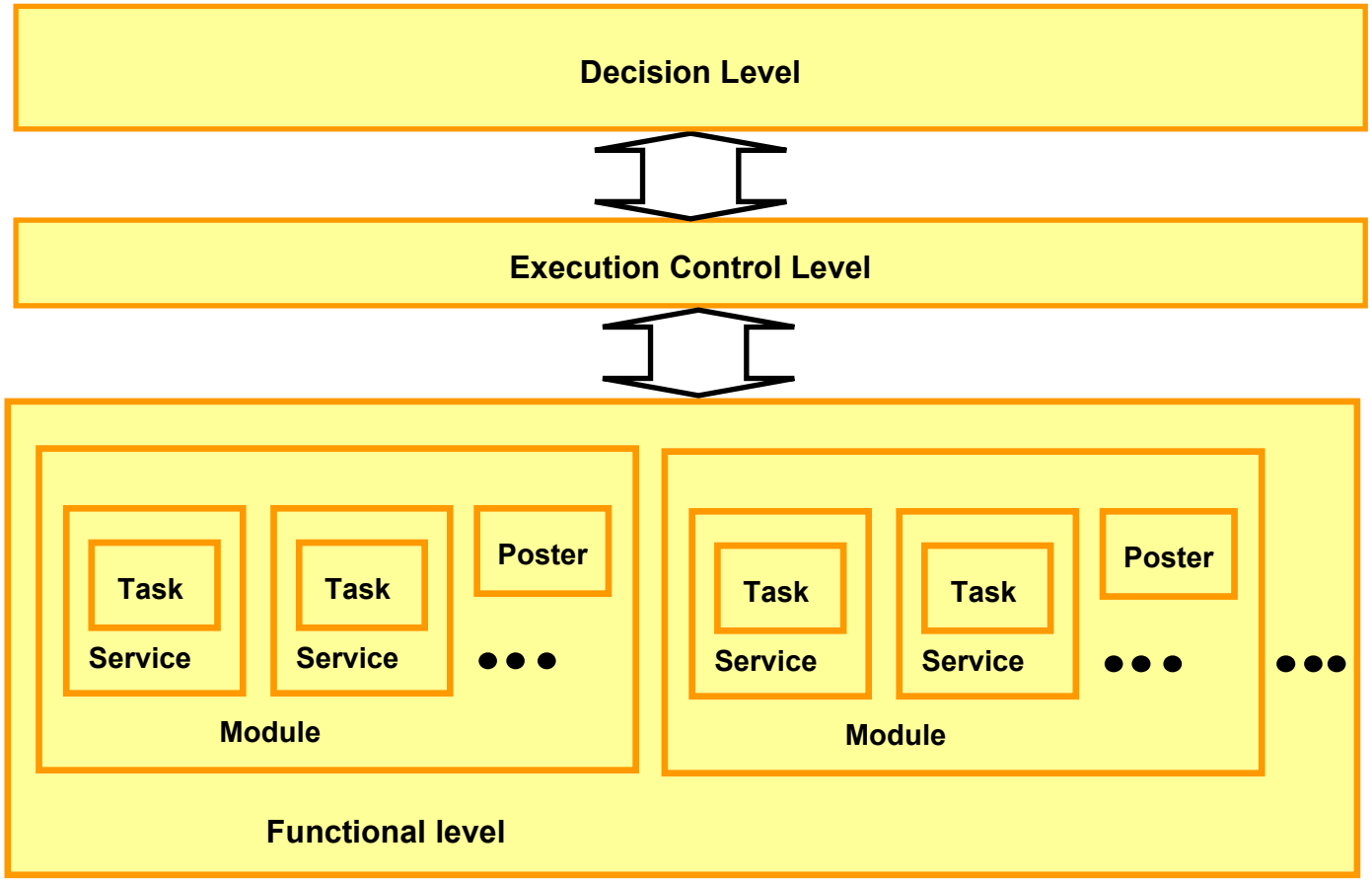


[Bliudze&Sifakis, Concur 08]

OVERVIEW

- Component-based Construction
- BIP: Basic Concepts
- Expressiveness
- Methodology for Componentization
- Discussion

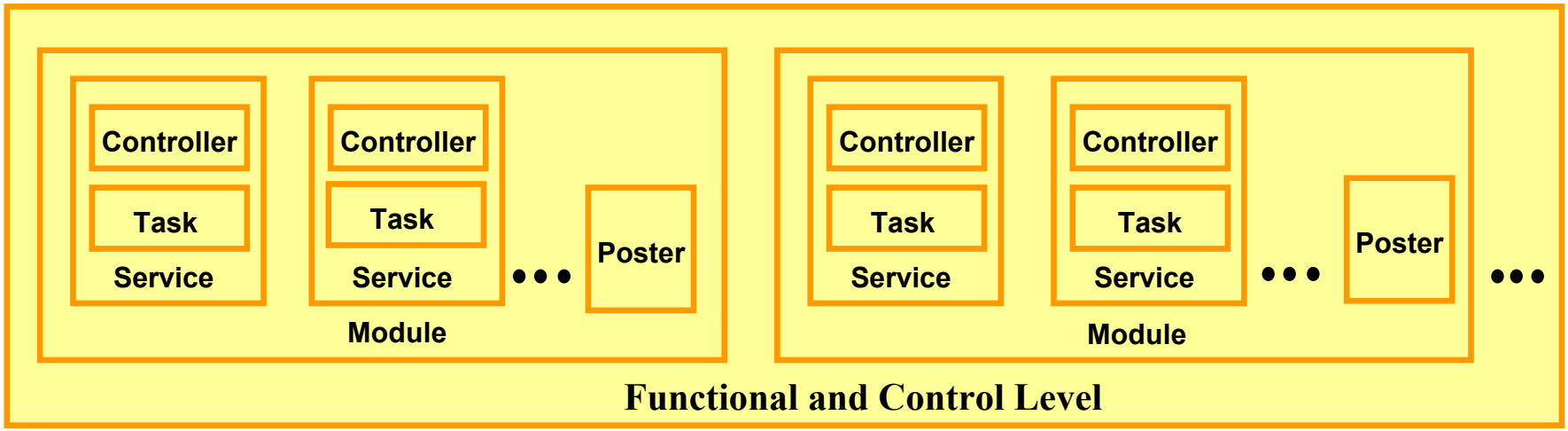
The DALA Robot: Architecture



Autonomous system of ~300000 lines of C++ code developed at LAAS laboratory in Toulouse

Objective: Design a componentized version where global Execution Control is replaced by Local Execution Control for each Service

The DALA Robot: Componentization



Functional and Control Level ::= Module^{+}

Module ::= Service^{+} . Poster

Service ::= Service Controller . Service Task

Service Controller ::= Event Triggered Controller | Cyclic Controller

Cyclic Controller ::= Event Triggered Controller . Cyclic Trigger

Service Task ::= Timed Task | Untimed Task

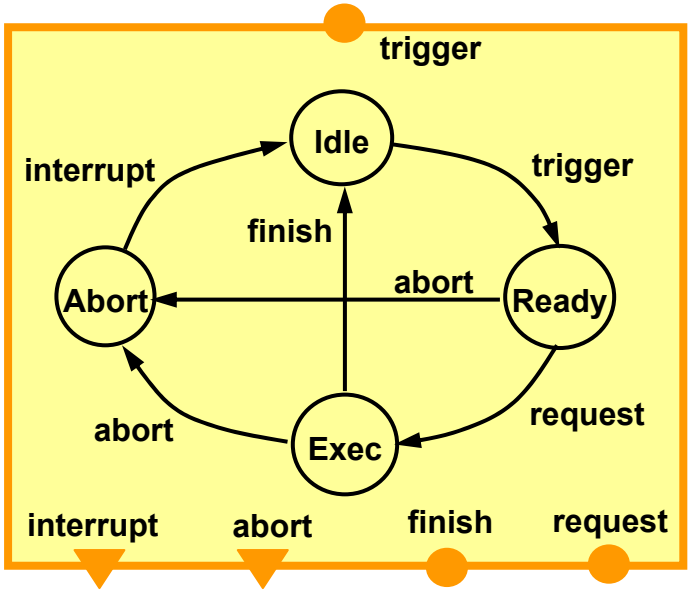
The DALA Robot: Event Triggered Controller

Idle: the Service is idle

Ready: checks the possibility for starting a new Task of the Service

Exec: execution of the Task of the Service

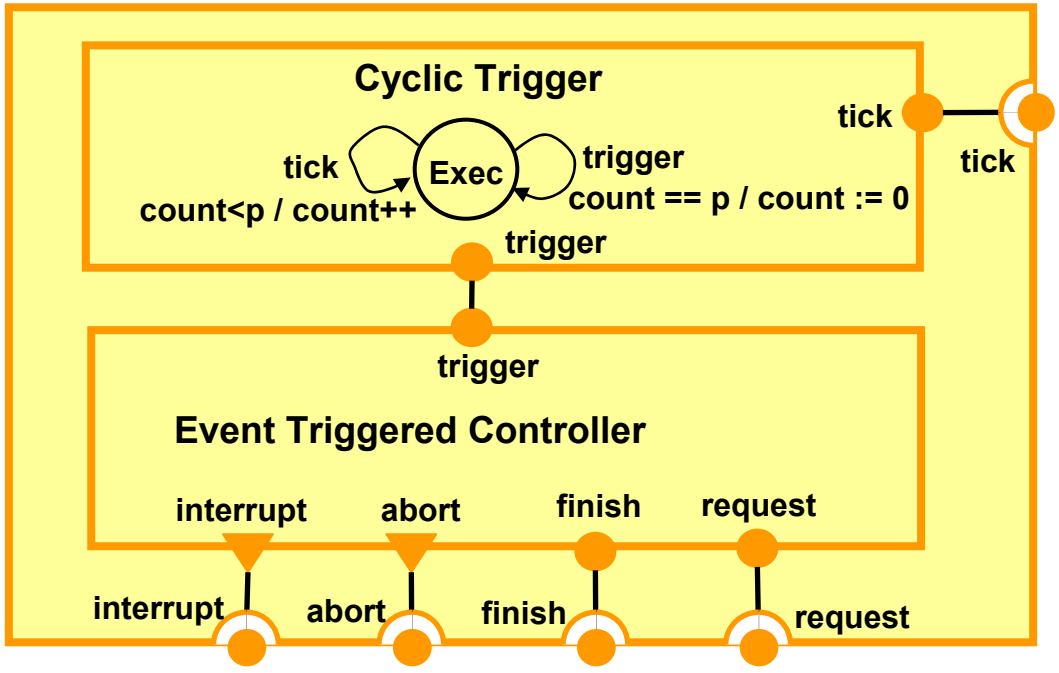
Abort: Service is aborted



The DALA Robot: Cyclic Controller

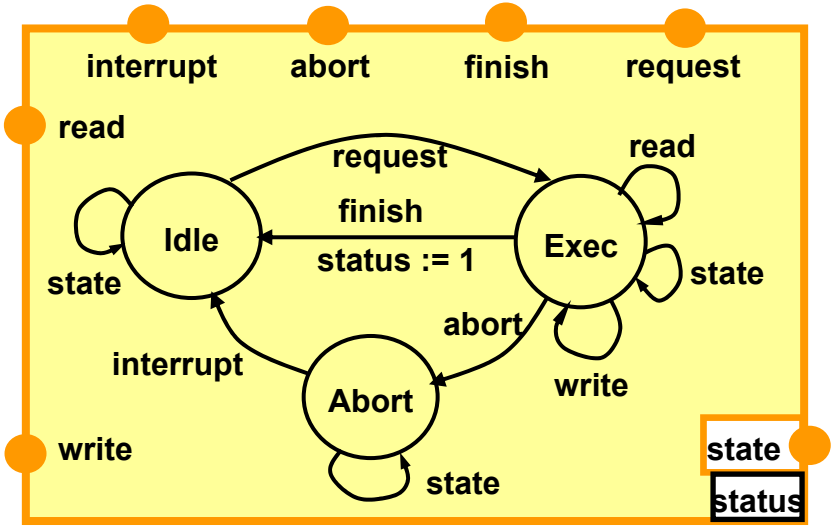
Cyclic Controller ::=
Event Triggered Controller . Cyclic Trigger

The Cyclic Trigger starts the Event Triggered Controller every period p



The DALA Robot: Untimed Task

Triggered by *request*



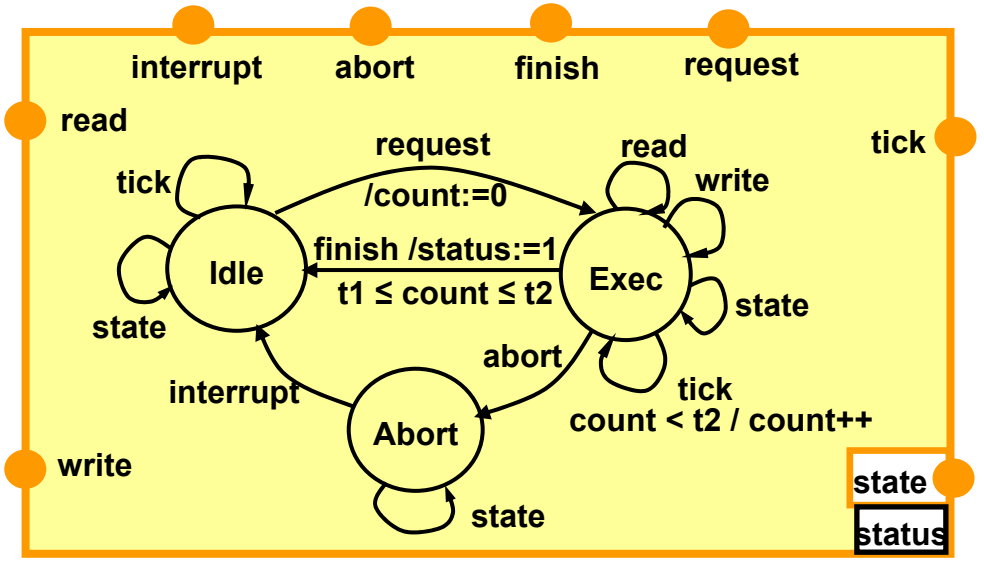
The variable ***status*** specifies the previous state of Task

status == 1 : Task successfully executed

status == 0 : Task aborted

The DALA Robot: Timed Task

- Obtained from an Untimed Task
- Its execution time is in [t1,t2]

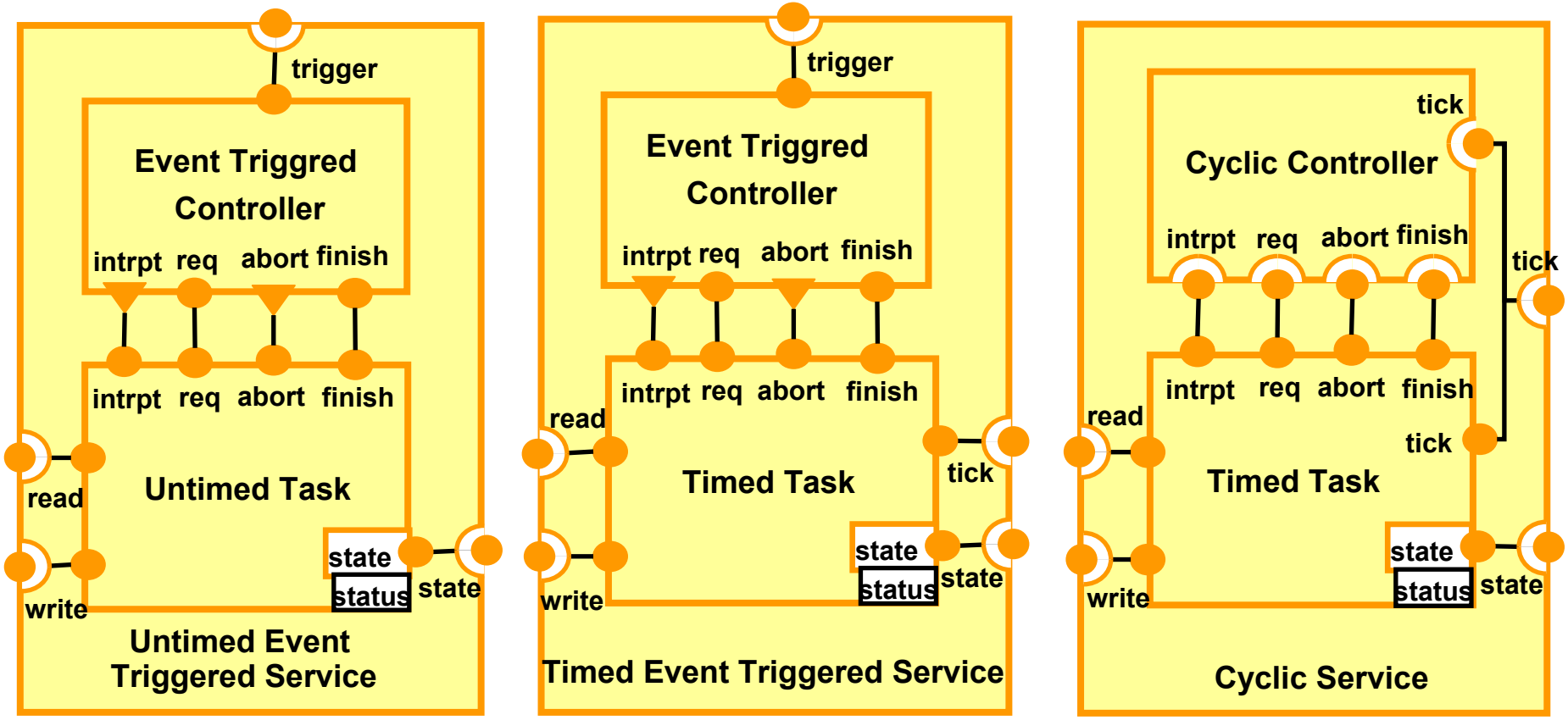


The DALA Robot: Different types of Services

Untimed Event Triggered Service
::= Event Triggered Controller. Untimed Task

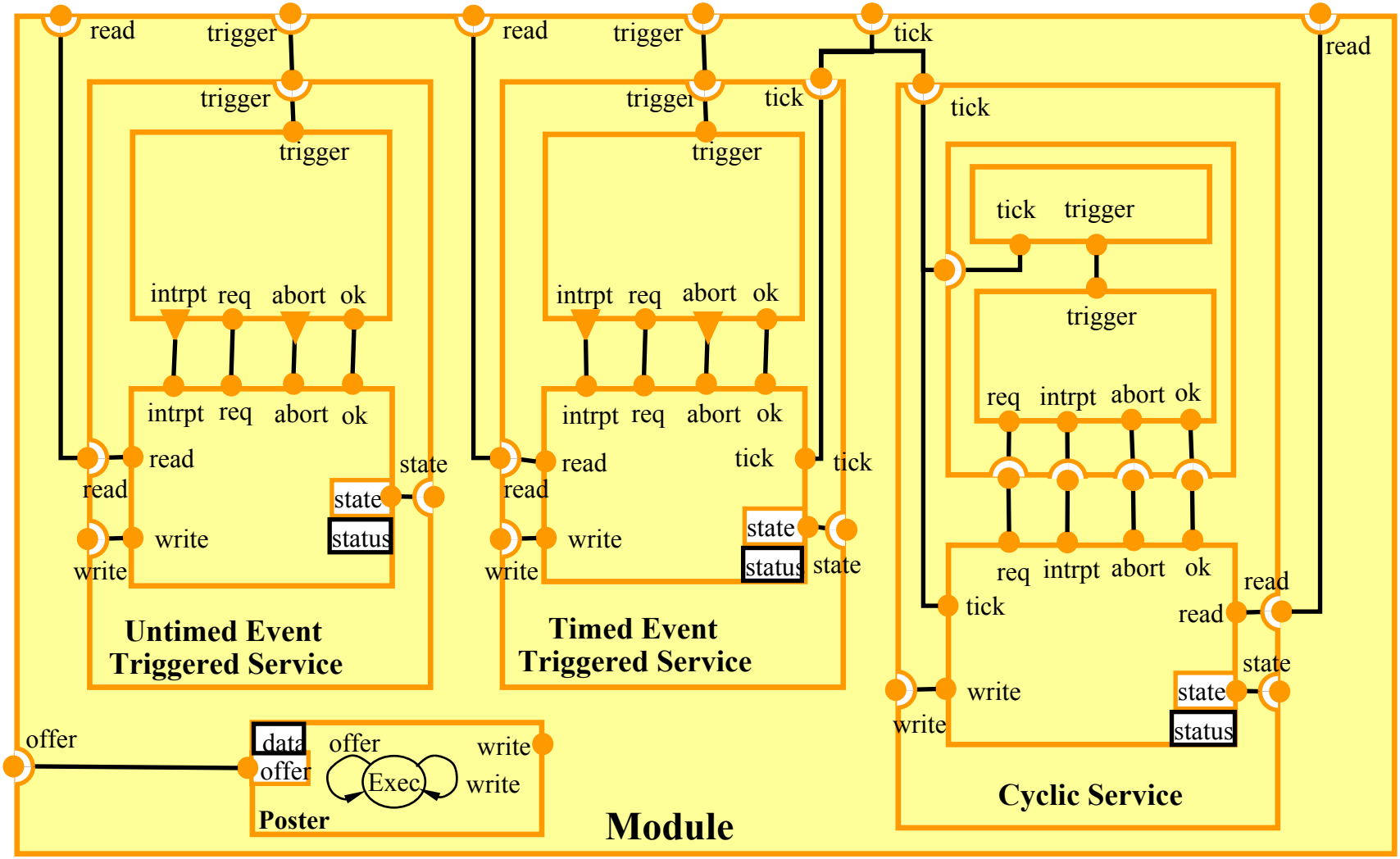
Timed Event Triggered Service ::= Event Triggered Controller. Timed Task

Cyclic Service ::= Cyclic Controller . Timed Task



The DALA Robot: A Module

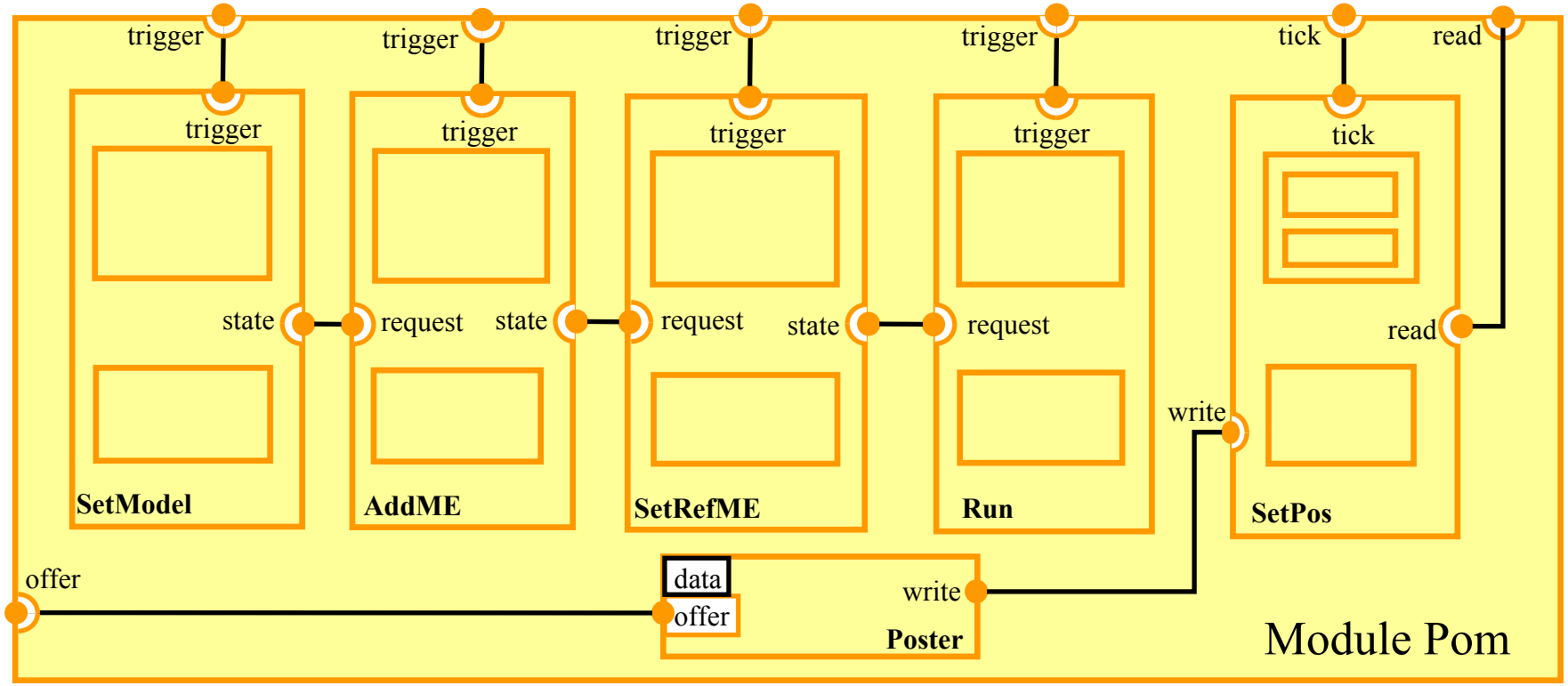
A module composed of 3 services and a poster



The DALA Robot: The module Pom

Reads and integrates data to provide an estimate of the position of the robot

Pom ::= SetModel. AddME. SetRefME. Run. SetPos. Poster

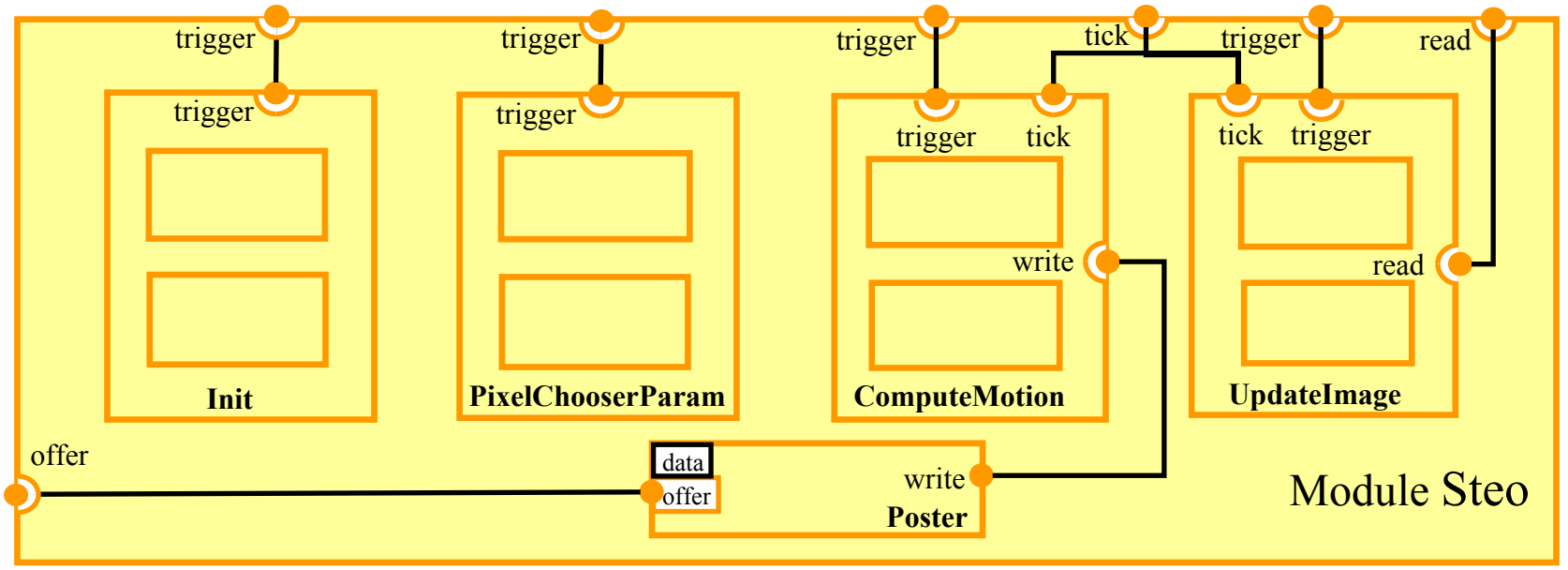


The DALA Robot: Module Steo

Computes the relative displacement of the robot

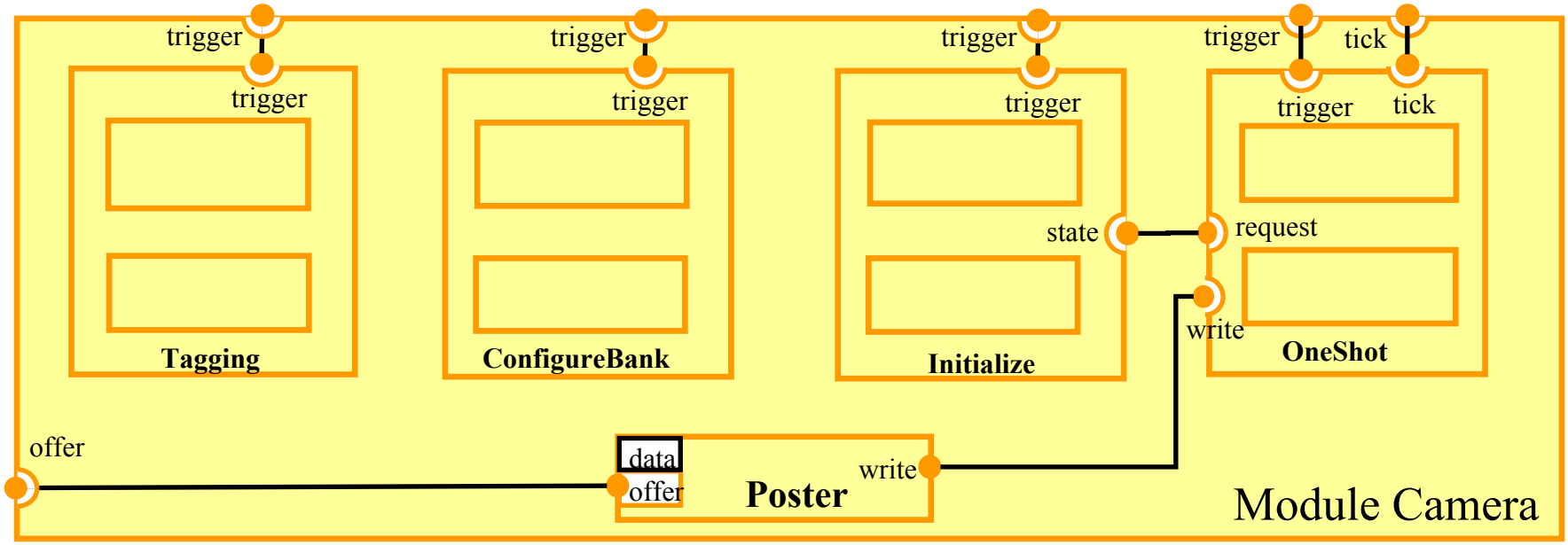
Steo ::=

Init. PixelChooserParam. ComputeMotion. UpdateImage. Poster



The DALA Robot: Module Camera

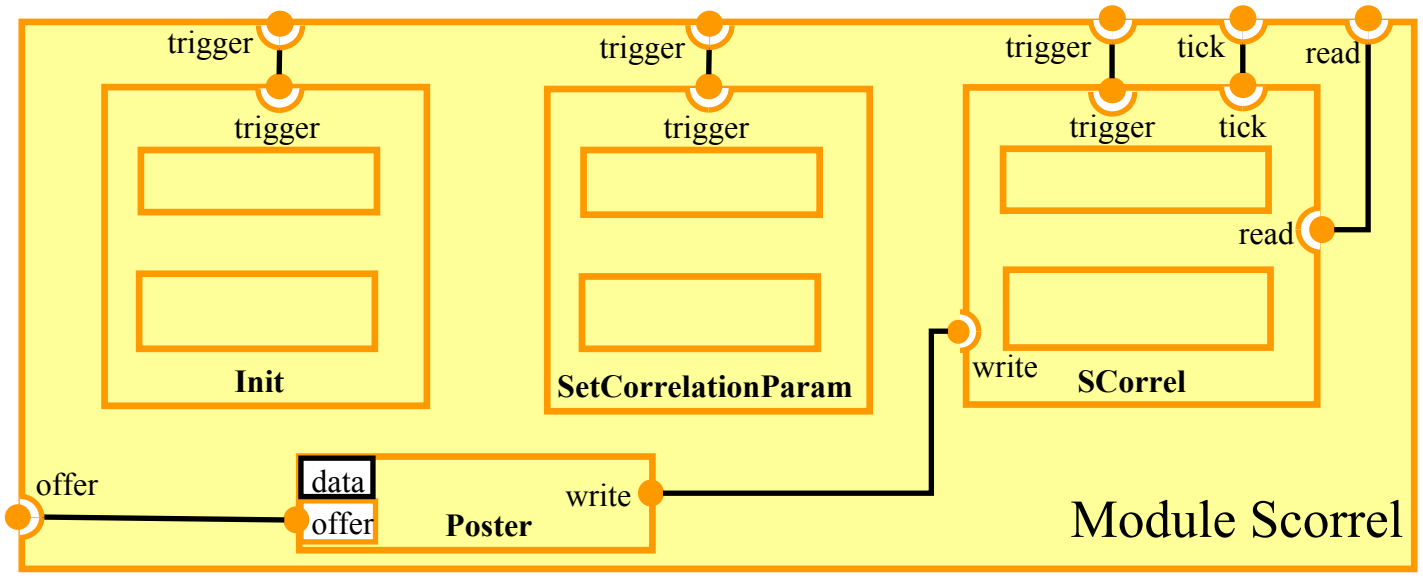
Camera ::= Tagging. ConfigureBank. Initialize. OneShot. Poster



The DALA Robot: Module Scorrel

Computes the stereo-correlation and produces 3D information about the environment.

Scorrel ::= Init. SetCorrelationParam. SCorrel. Poster



Discussion

Principles

- transitions of composite components are obtained as the interaction of transitions of their constituents
- strict separation between behavior and composition operators e.g. no expansion theorems expressing composition in terms of non-deterministic choice and prefixing by actions.

Work in progress

- Not yet completely explored possible relations between glues of process calculi. However, these cannot be as expressive as glues with negative premises even by allowing additional behavior for coordination
- We considered equivalences where all the ports are observable. The preservation of the presented results for observational relations should be investigated.

Thank You