Politechnika Wrocławska

Event Condition Action Approach to Process' Control Layer Modeling in Unified Process Metamodel

5th International Conference on Systems and Informatics (ICSAI 2018)

Krystian Wojtkiewicz

Department of Information Systems
Faculty of Computer Science and Management
Wrocław University of Science and Technology

11/11/2018

HR EXCELLENCE IN RESEARCH

Politechnika Wrocławska

Agenda

Introduction

Resources in UPM
- Static resources
- Flowing resources
- Resource flow

Control Layer

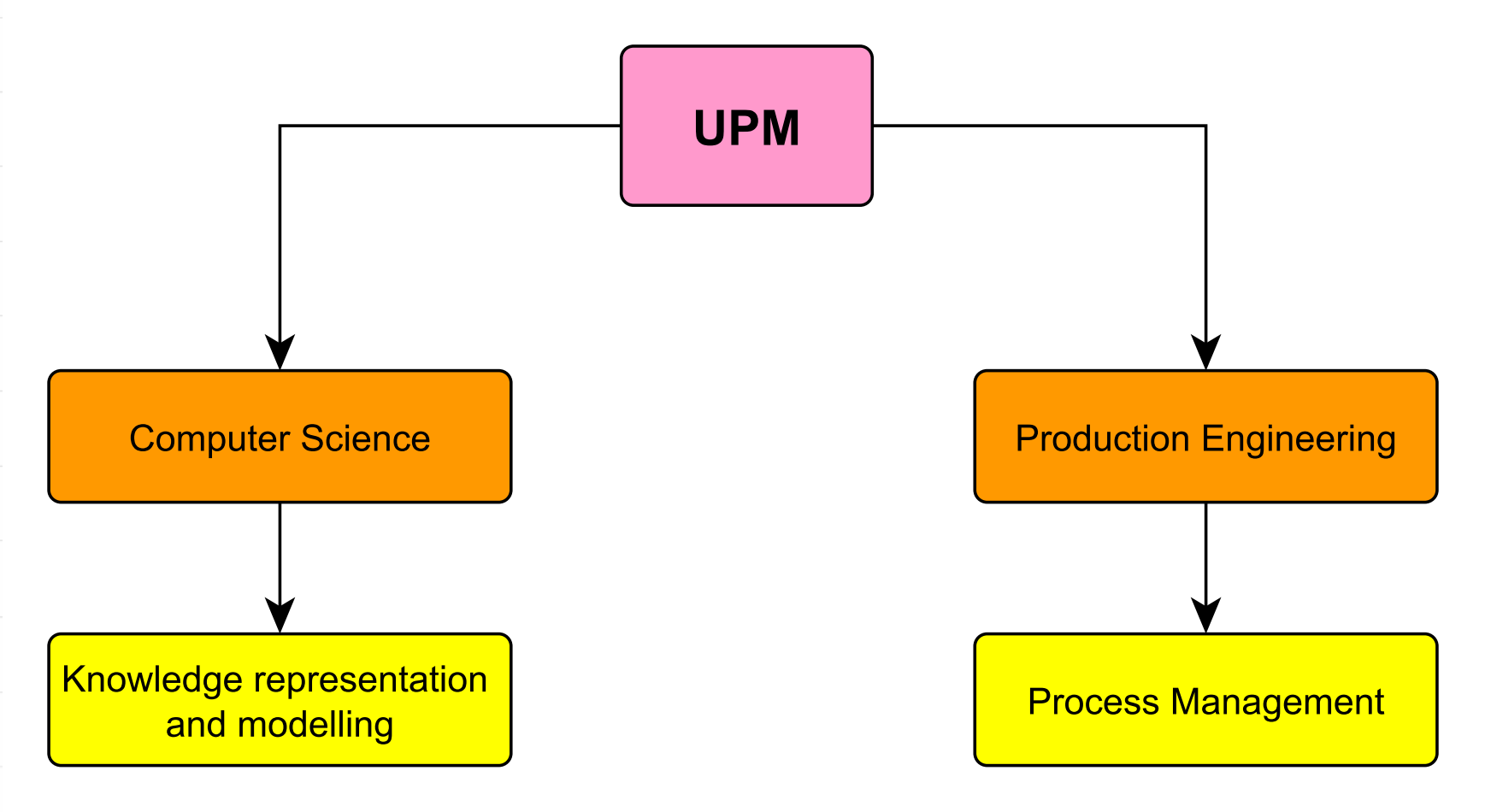
Event–Condition–Action

Conclusions

Politechnika Wrocławska

Introduction

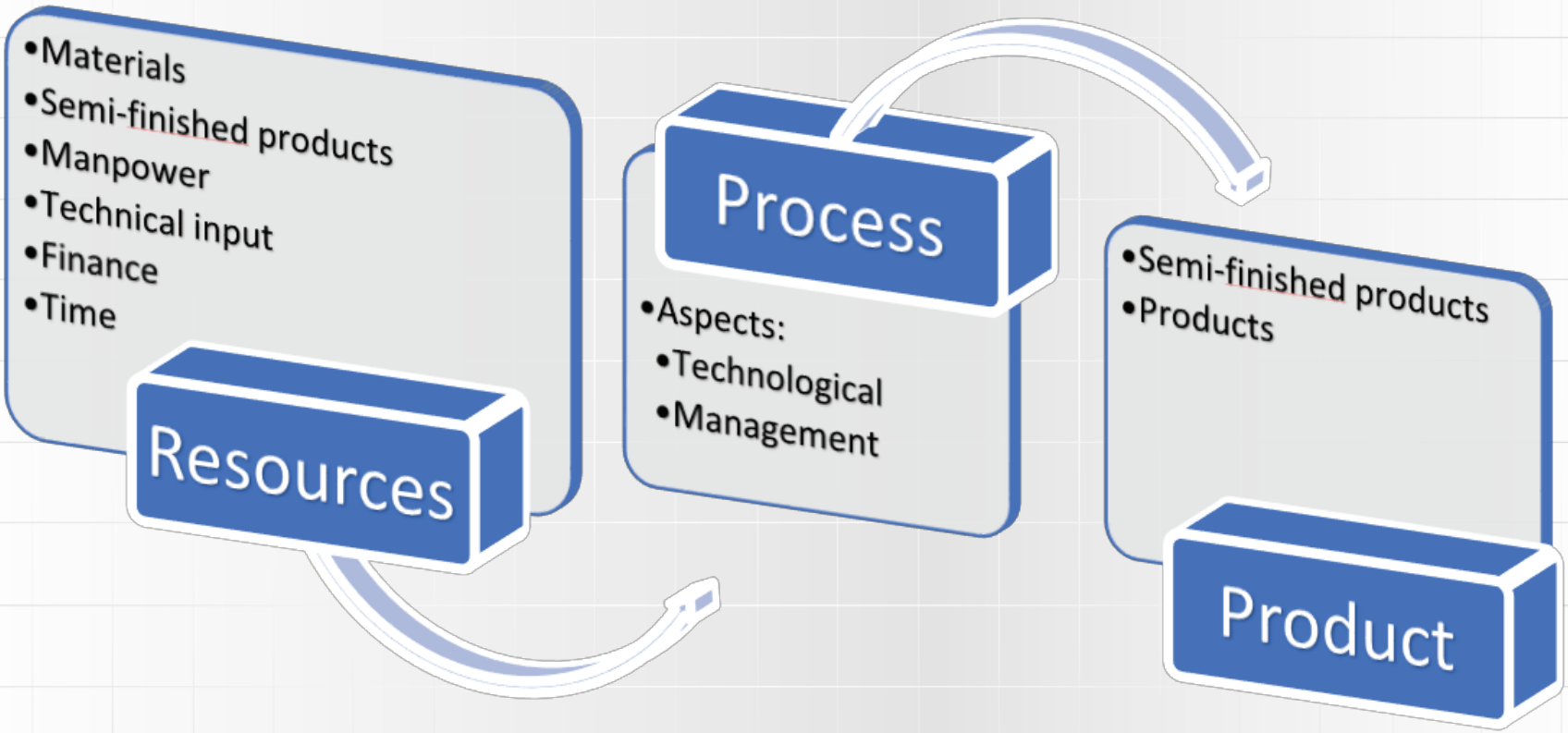
Research field



Politechnika Wrocławska

Introduction

What is a process?



Introduction

UPM – Unified Process Metamodel

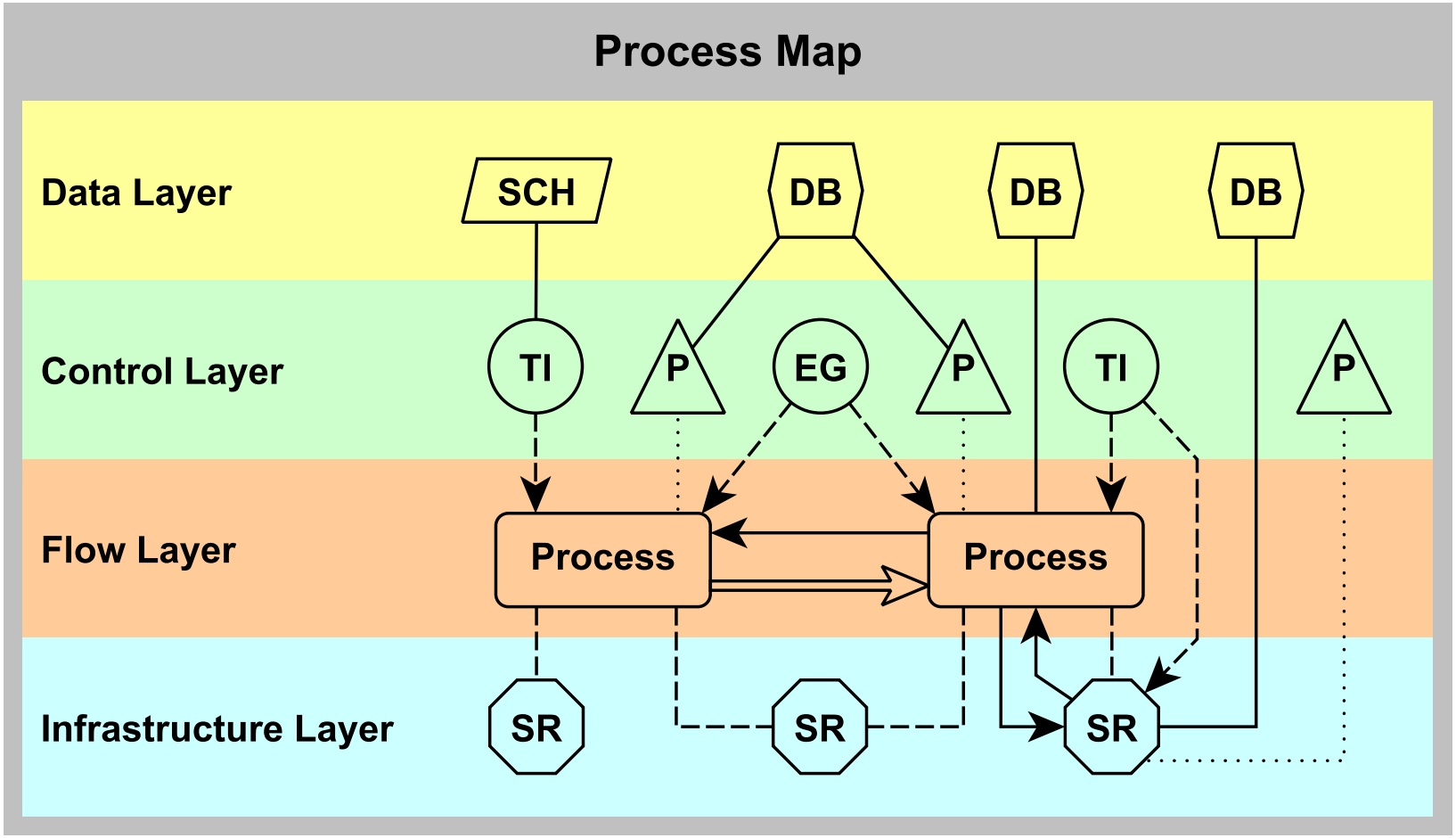
The Unified Process Metamodel (UPM) consists of:

- theoretical base (full identification of categories and relationships among them),
- formal notation,
- modelling language,
- algorithms and processing methods.

Politechnika Wrocławska

Introduction

UPM – Modelling Layers



Resources in UPM

UPM – Yet another approach to resources definition

There are two aspects of the resource management in *UPM*:

- *Static resources*
- *Flowing resources*

Politechnika Wrocławska

Resources in UPM

Static resources

UPM – Static Resource

- The Static Resource is the sole component of the infrastructure layer that is responsible for making resources available.
- The Static Resource is not subject to processing, flows or other operations.
- It is the base for the components of the *resource source* and the *resource target* so that they could generate or absorb flowing resources.

$$\sigma(SR) = (id : ID, conn : \langle C \rangle) \tag{1}$$

where:

SR – type defining the static resource,

id – component identifier,

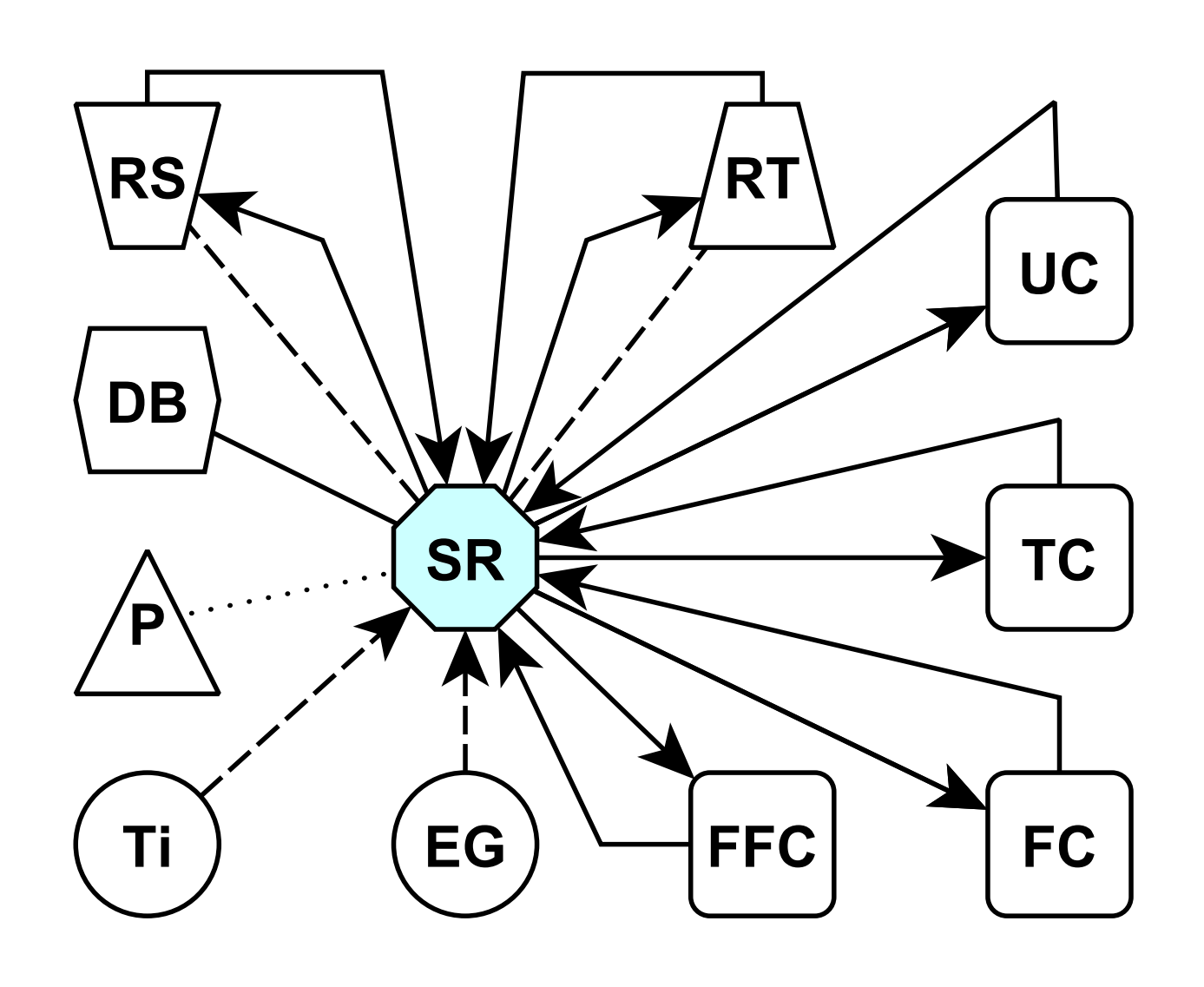
$conn$ – list of connectors of a given component.

Politechnika Wrocławska

Resources in UPM

Static resources

UPM – Static Resource



Politechnika Wrocławska

Resources in UPM

Flowing resources

UPM – Flowing Resource

The *Flowing Resource (FR)* introduced by the UPM should be understood as material or immaterial being having discrete or continuous nature and which is an input or output element of the process

FR is generated in *Resource Source* that utilize *Static Resource*. FR is absorbed by dedicated *Resource Target*connected with *Static Resource*.

$$\sigma(FR) = (id : ID, unit : ID, pack : double, storage : bool) \quad (2)$$

where:

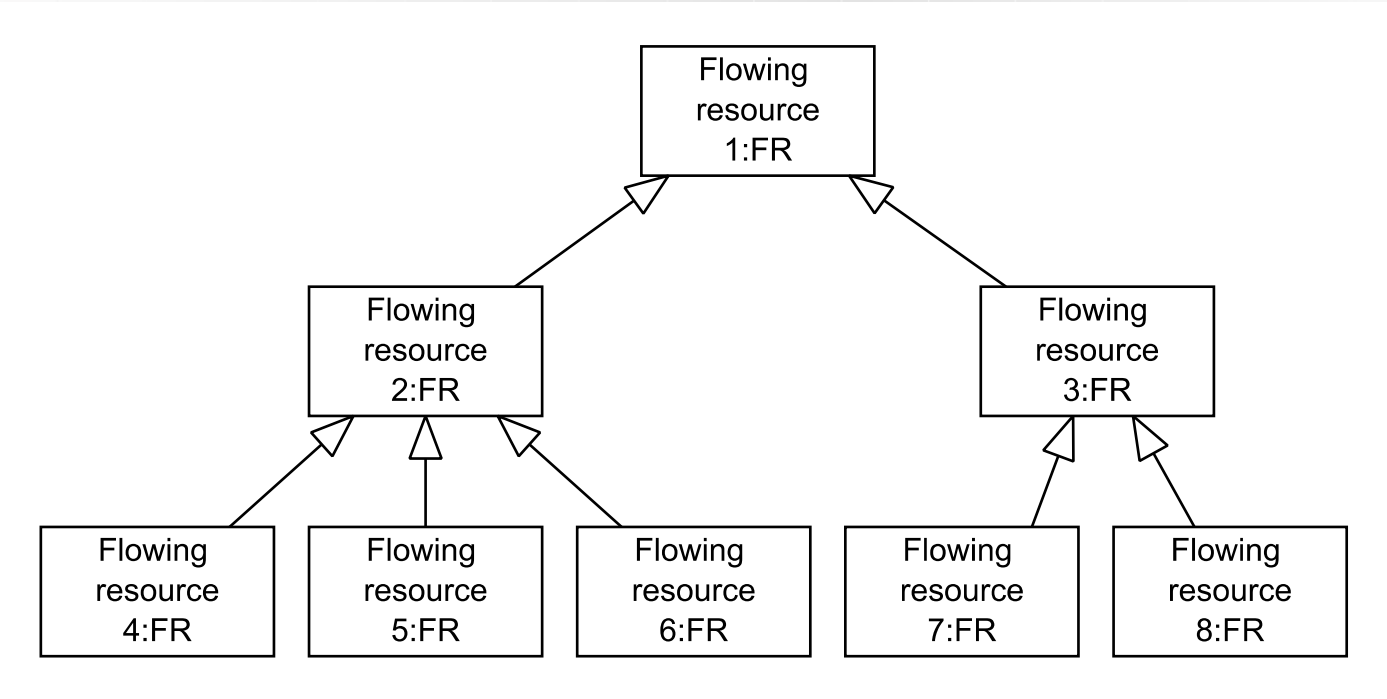
- *FR* – *flowing resource* type,
- *id* – *flowing resource* identifier,
- *unit* – basic measurement unit of the *flowing resource*,
- *pack* – the smallest amount ("pack") of the resource which can be used,
- *storage* – Boolean value determining whether a given *flowing resource* has the ability to store it

Politechnika Wrocławska

Resources in UPM

Flowing resources

UPM – Flowing Resource exemplary taxonomy



Politechnika Wrocławska

Resource flow control possibilities:

- neutral - no change introduced,
- suction nature for inputs (In) – suction pump (SP),
- forcing nature for outputs (Out) – force pump (FP).

We measure it by the means of resource flow force:

$$rfr = \frac{fr}{t} \tag{3}$$

where:

rfr – *resource flow rate*,

fr – *flowing resource amount*,

t – *time*.

Politechnika Wrocławska

Resources in UPM

Resource flow

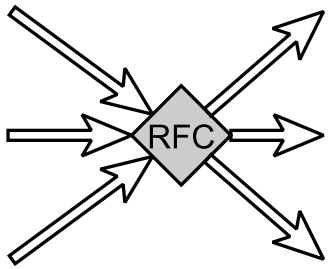
Resource Flow Control module

The Resource Flow Control module, as a type, is formally defined as:

$$\sigma(RFC) = \left\langle \begin{array}{l} id : ID, \\ in : \langle PORT \rangle, \\ out : \langle PORT \rangle, \\ sep : \langle double \rangle \end{array} \right\rangle; \tag{4}$$

where:

- id – module identifier,
- in – list of input ports,
- out – list of output ports,
- sep – list of resource flow separation coefficient.



Politechnika Wrocławska

Resources in UPM

Resource flow

Types of Resource Flow Control

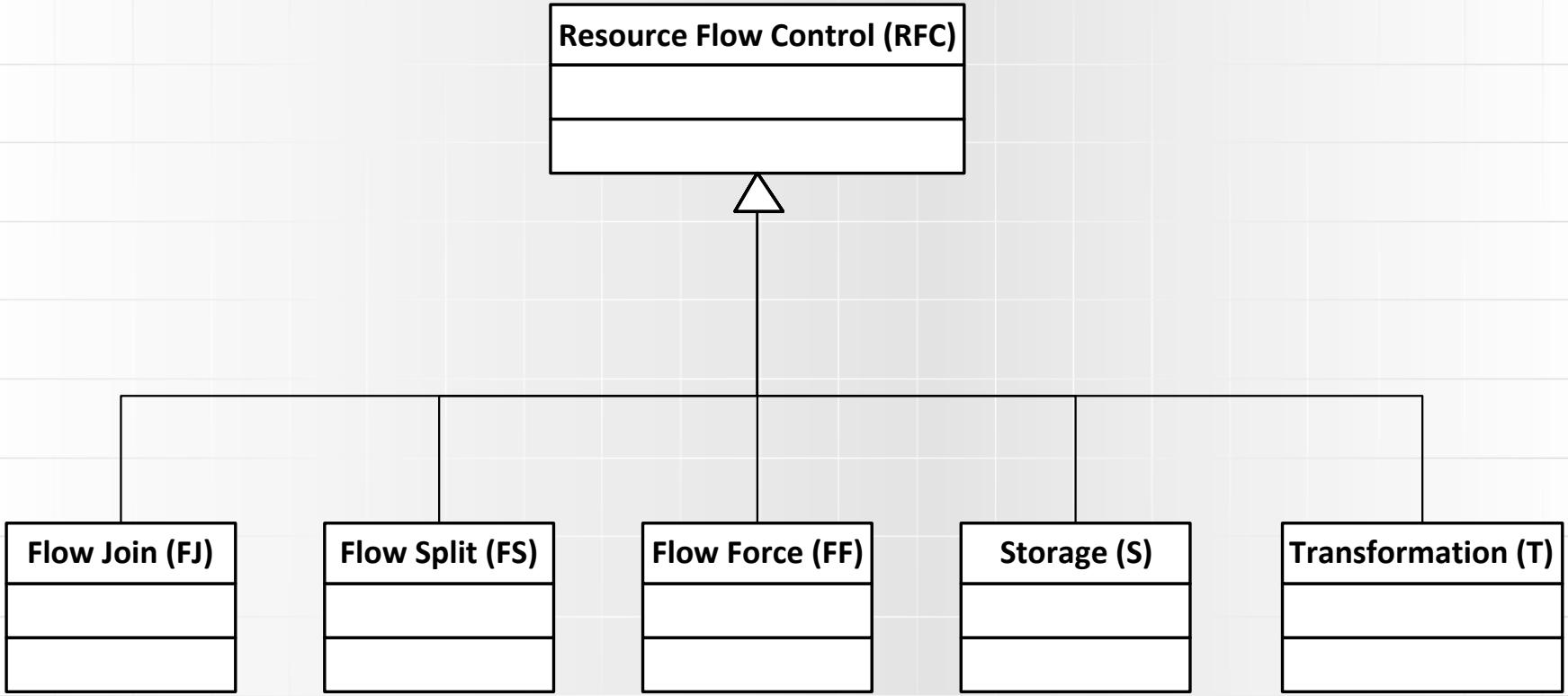


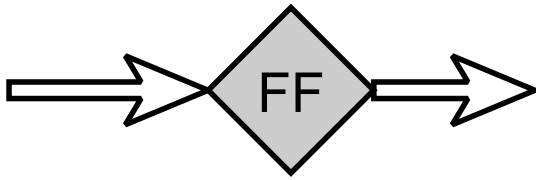
Figure: Taxonomy of Resource Flow Control Component

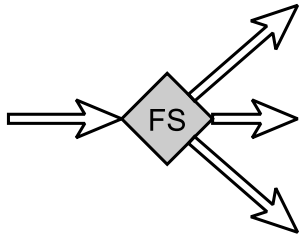
Politechnika Wrocławska

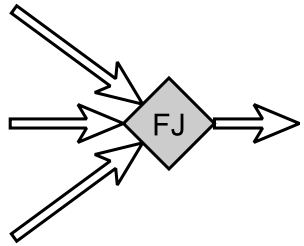
Resources in UPM

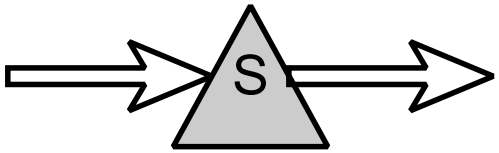
Resource flow

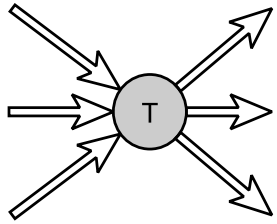
Types of Resource Flow Control

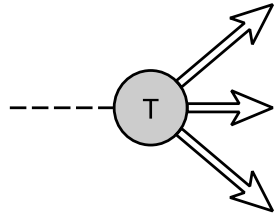












Politechnika Wrocławska

Formally *control layer* as a type is defined as:

$$\sigma(CL) = (ti : \langle TI \rangle, eg : \langle EG \rangle, p : \langle P \rangle) \tag{5}$$

where:

CL – type defining *control layer*,

TI – type defining *timer* component,

EG – type defining *event generator* component,

P – type defining *procedure* component,

ti – list of *timer* components,

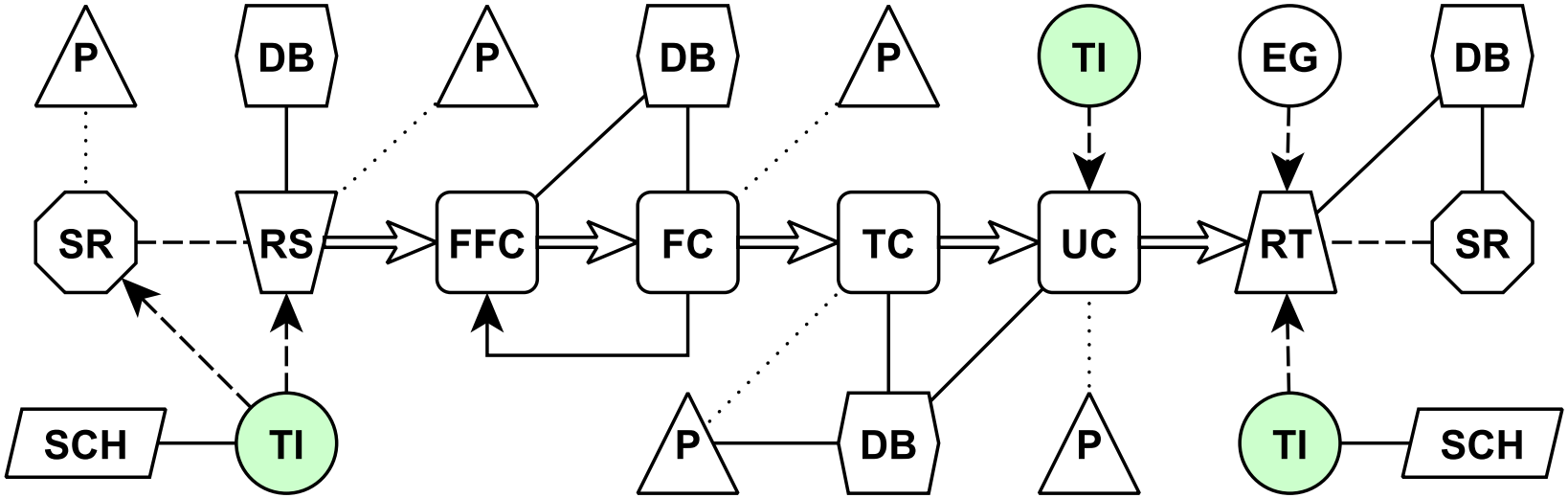
eg – list *event generator* components,

p – list *procedures* components.

Politechnika Wrocławska

Control Layer

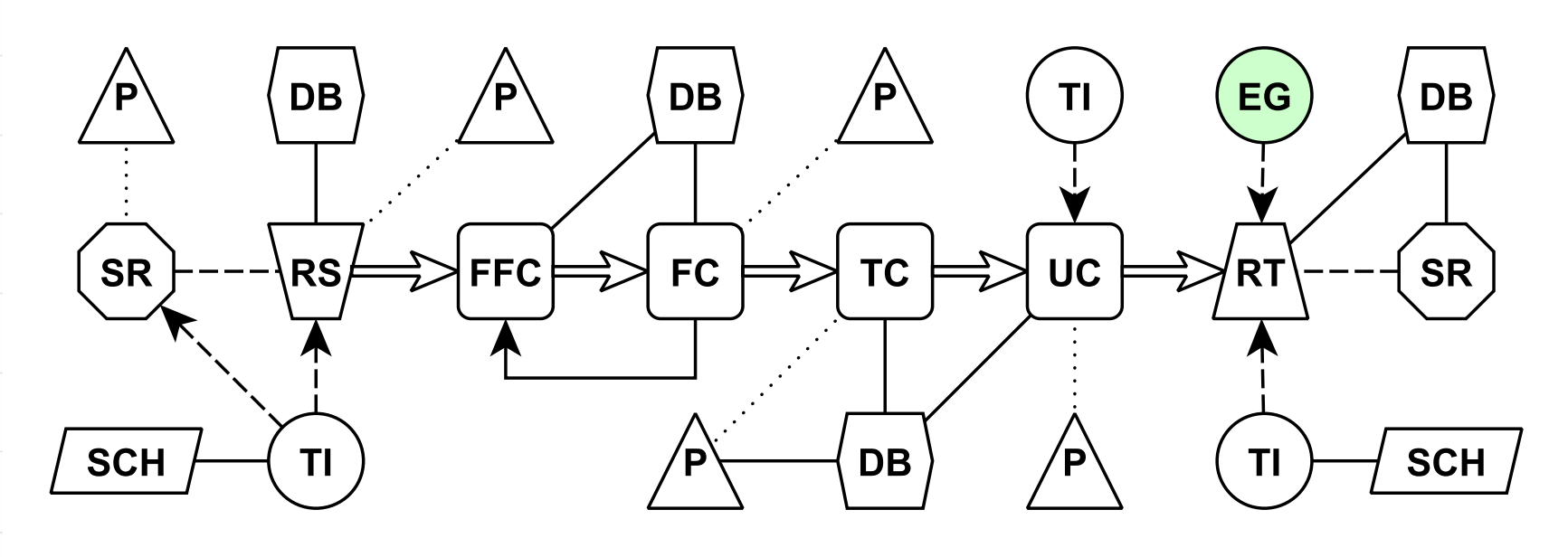
Timer



Politechnika Wrocławska

Control Layer

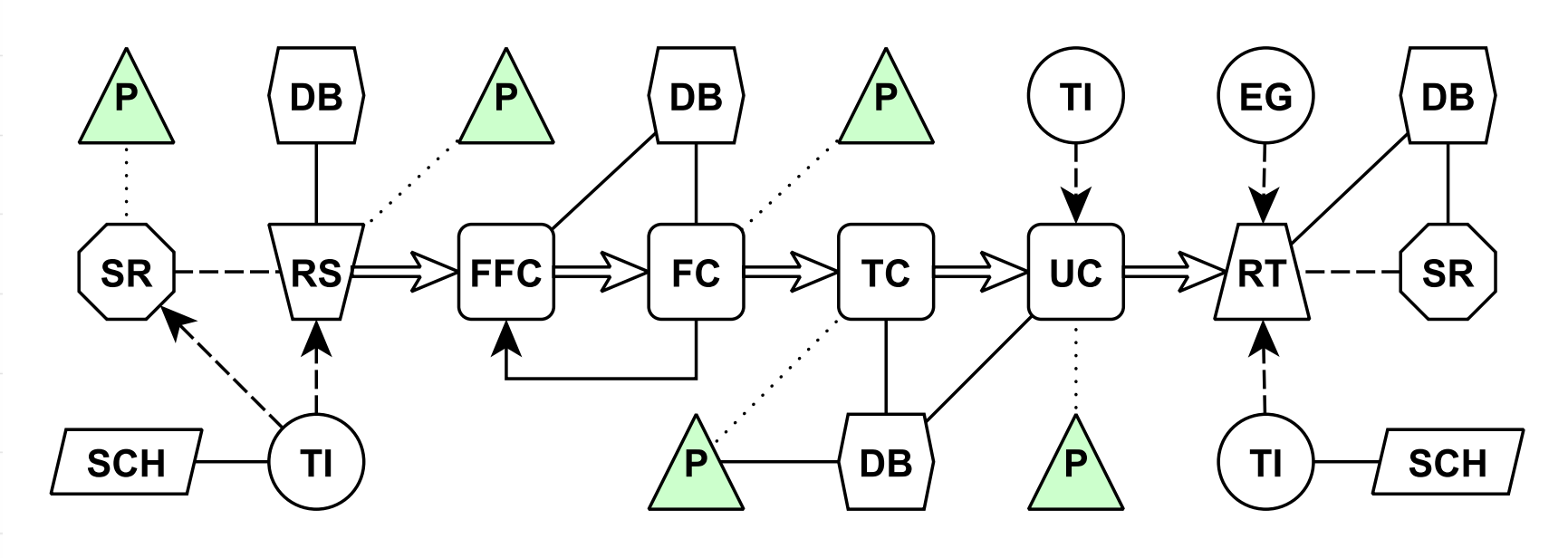
Event generator



Politechnika Wrocławska

Control Layer

Procedures



Politechnika Wrocławska

Event–Condition–Action

Concept of ECA

Event Condition Action (ECA) concept is based on the assumption that:

- *Event* (EV) is a signal initializing the *condition* fulfillment verification,
- *Condition* (CN) is a logic element, that if checked to be *true* triggers the *action*,
- *Action* (AC) is a manifestation of *script* that is assigned to the *event*.

Event–Condition–Action

Triggers of ECA

The sequence based on *message* sending mechanism is defined as follows:

$$\textit{Message}\ (\mathrm{M}) \mapsto \textit{Event}\ (\mathrm{EV}) \mapsto \textit{Condition}\ (\mathrm{CN}) \mapsto \textit{Action}\ (\mathrm{AC}),$$

in case of *events* generated by components

$$\textit{Timer}\ (\mathrm{TI}) \mapsto \textit{Event}\ (\mathrm{EV}) \mapsto \textit{Condition}\ (\mathrm{CN}) \mapsto \textit{Action}\ (\mathrm{AC}),$$

or

$$\textit{Event Generator}\ (\mathrm{EG}) \mapsto \textit{Event}\ (\mathrm{EV}) \mapsto \textit{Condition}\ (\mathrm{CN}) \mapsto \textit{Action}\ (\mathrm{AC}).$$

Politechnika Wrocławska

Conclusions

Conclusions and future work

1. The new approach towards process definition has been proposed
2. The notion of resource in the process has been unambiguously defined
3. Even–Condition–Action schema has been applied to control layer default operation

Future work:

1. Implementation of design tool for process definition
2. Implementation of simulation environment

Politechnika Wrocławska

Thank you for your attention

Krystian Wojtkiewicz
krystian.wojtkiewicz@pwr.edu.pl

homepage	i-cu.eu
skype	skokrys
researcherID	K-6939-2017
ORCID	0000-0002-7851-4330
ResearchGate	Krystian_Wojtkiewicz2