Lecture
Organic Computing II
Summer term 2019

Chapter 4: Observer/Controller Architecture

Lecturer: Jörg Hähner

Agenda

Content

- General design concept for organic systems
- Observer/Controller architecture
- Multi-level Observer/Controller framework
- Conclusion and further readings

Goals

Students should be able to:

- Outline the basic design concept used in OC.
- Sketch the Observer/Controller approach and the multi-level extension.
- Compare distribution variants.
- Explain the responsibilities of the contained components.

Agenda

- General design concept for organic systems

- Observer/Controller architecture

- Multi-level Observer/Controller framework

- Conclusion and further readings
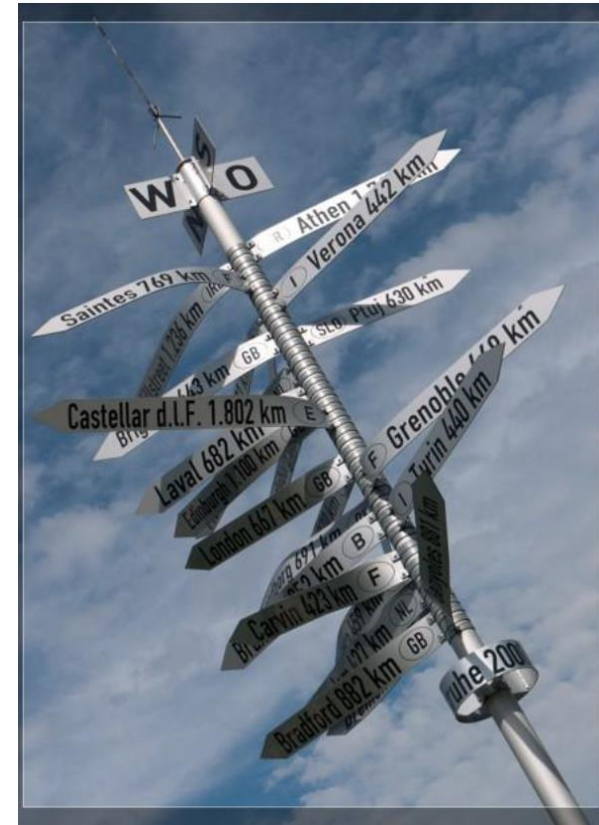
General concept of an organic system
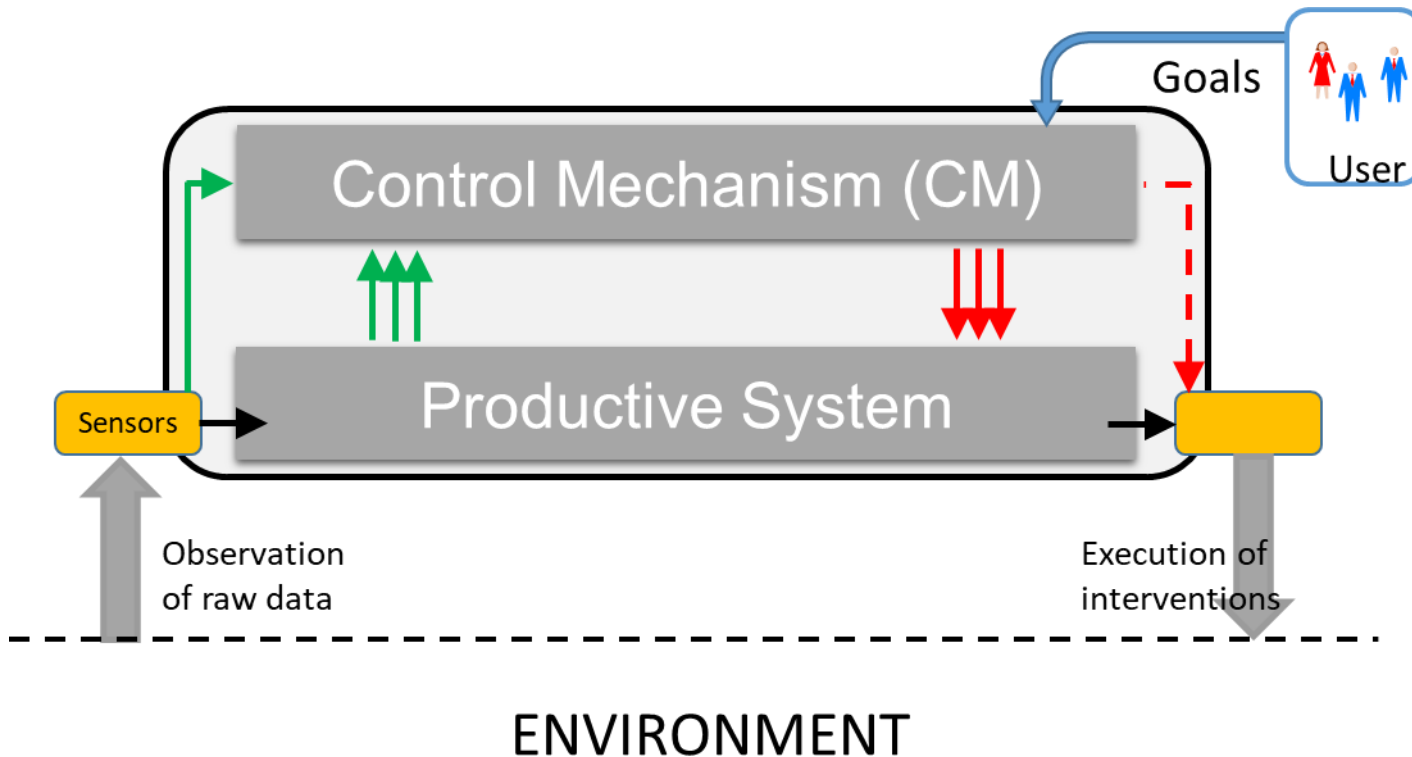
Green: flow of observed data
Red: flow of control interventions

System boundaries

- Everything not directly related to the system itself is modelled as being part of the external environment.

- Environment is accessed through sensors and manipulated by actuators.

Take the human operator "out of the loop"

- Productive system is only influenced by internal control mechanism.

- Theoretical possibility for the user to intervene directly with the output signal.

- User provides just a goal (or a set of goals) defining good or bad behaviour.

- CM figures out what to do in which situation without direct human intervention.

- Classic approach: user specifies directly what the system has to do.

General concept of an organic system (3)

Requirements

- Access to sensor readings covering all important aspects.

- Access to internal status variables of productive system.

- Utility function specifying good/bad behaviour.

- Access to control interfaces of productive system.

- Control interface has to be directly related to performance.

Actions

- Manipulation of parameters to guide behaviour of productive system.

- Activation or deactivation of components and functionality.

- Exchange of algorithms or techniques in use.

- Selection of interaction partners.

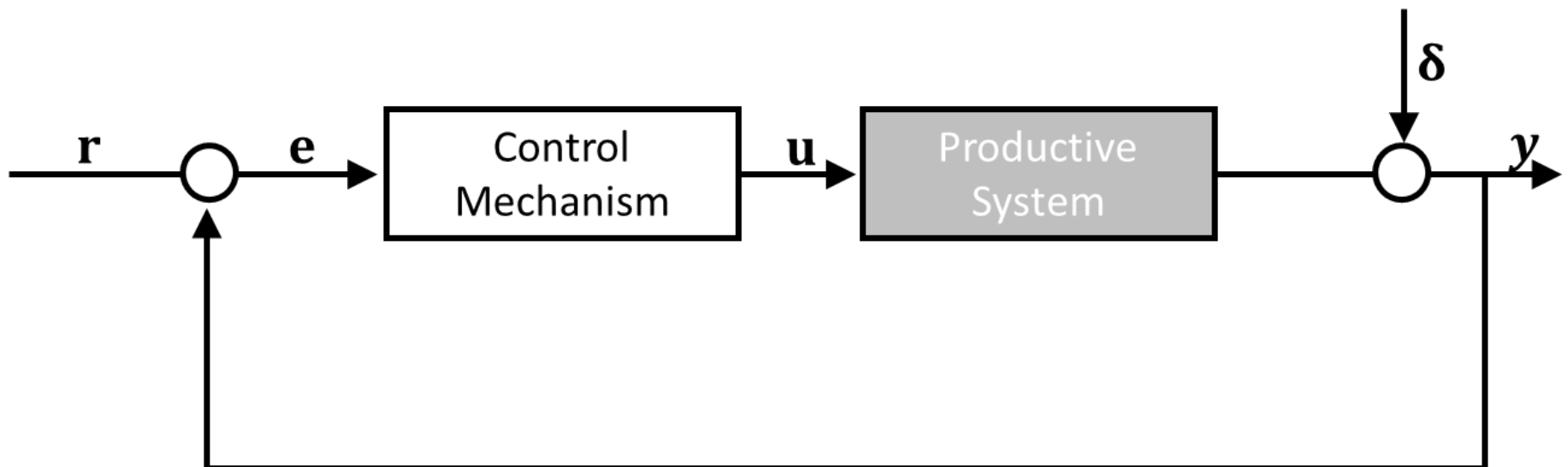- Modification of observation model (i.e. selection, resolution, and frequency of sensor information)

Example: Traffic control

- Productive system: traffic light controller

- Sensor readings: induction loop readings (occupied or not)

- Effectors: traffic signals (green/yellow/red)

- Input CM: vector with traffic flow per hour for each turning

- Output CM: green durations for each traffic light

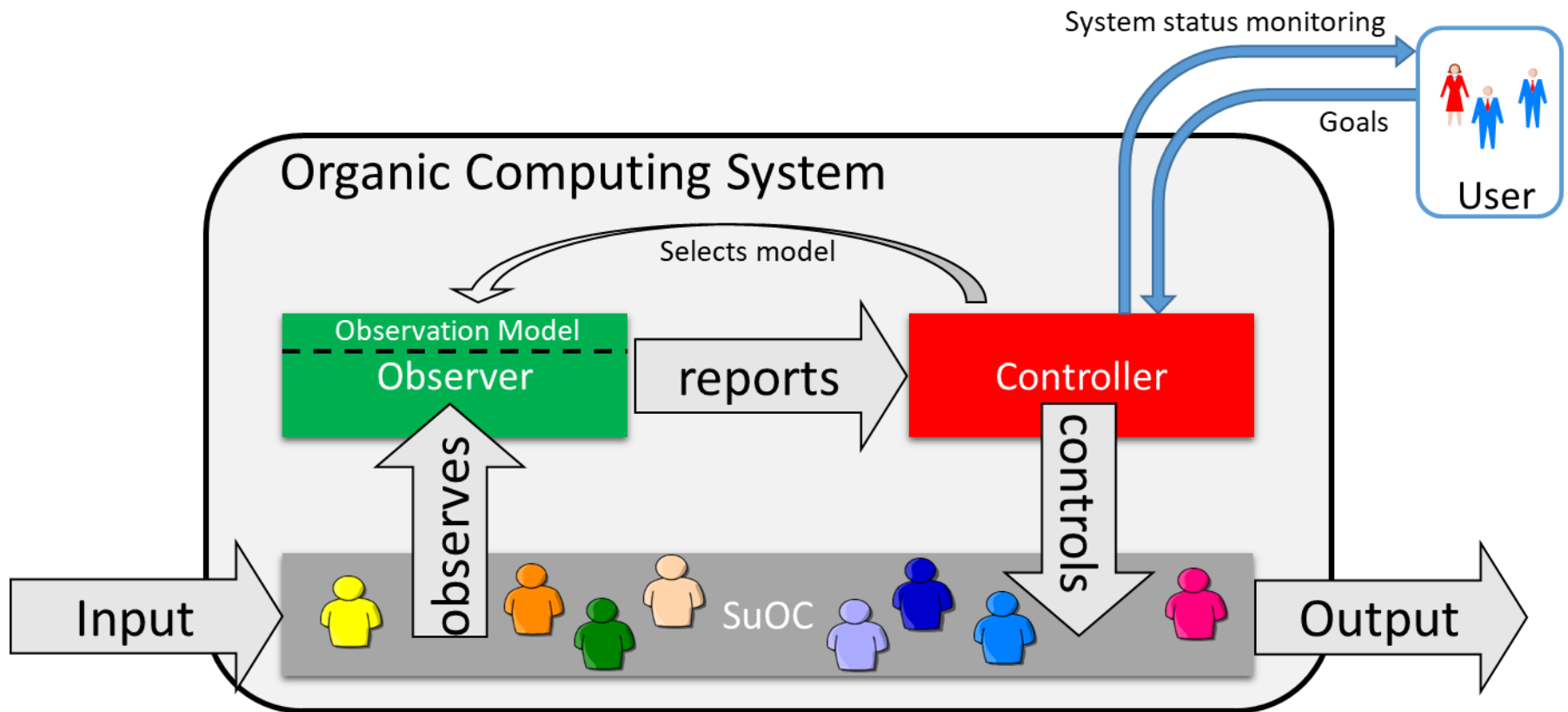# *General concept of an organic system (5)*
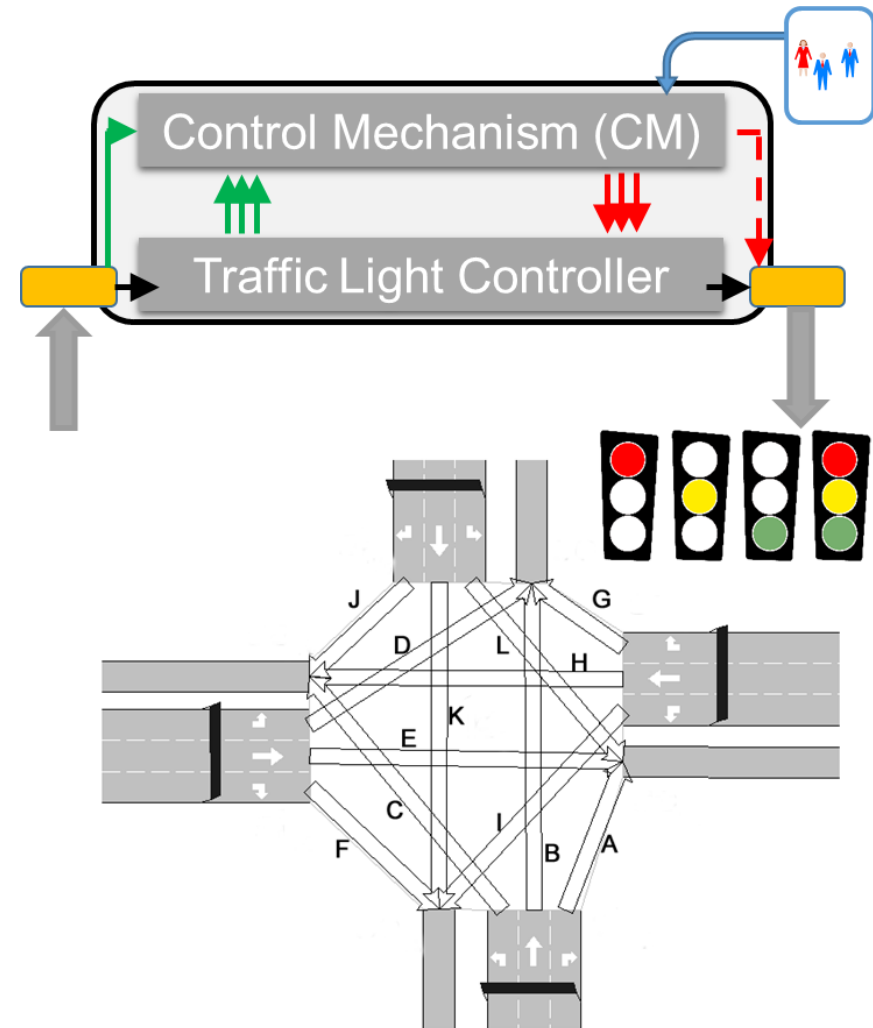
Control loop

- Actions have impact on performance

- Performance is measured from the environment

- Environment is modified by actions
  → Feedback loop (see control theory)

Agenda

- General design concept for organic systems

- Observer/Controller architecture

- Multi-level Observer/Controller framework

- Conclusion and further readings

Generic observer/controller pattern (2)

Three major components:

- **System under Observation and Control** (SuOC):

  – Productive part of the system

  – Functional even if higher-layered observer/controller components fail

  – Not restricted to individual devices

- **Observer:**

  – Gathers information

  – Analysis, predicts, detects patterns

  – Builds situation description

  – Based on observation model

- **Controller:**

  – Decides about actions

  – Learns from feedback

  – Modifies observation model

SuOC

Observer

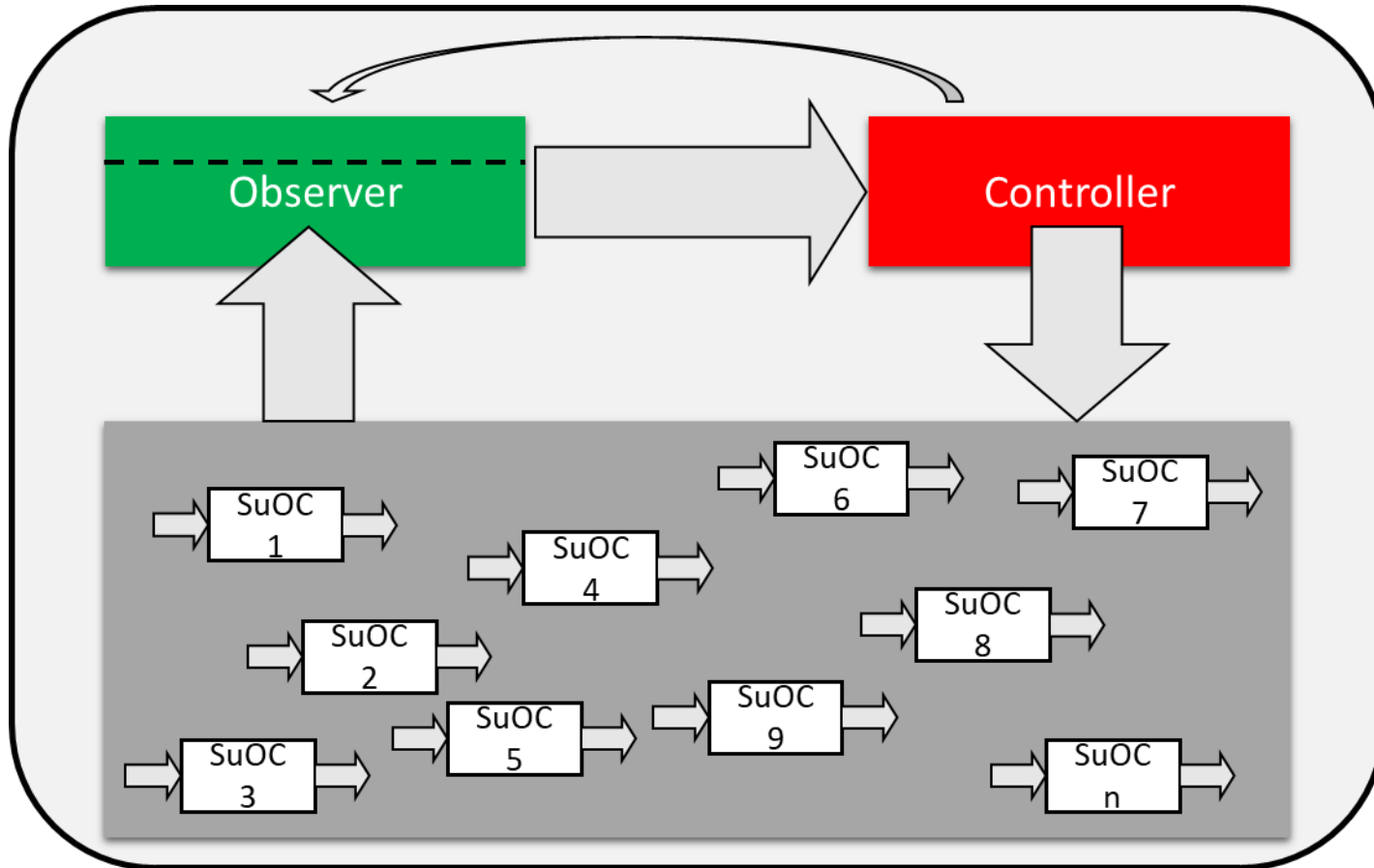Controller

Generic observer/controller pattern (3)

Learning from feedback

- Rule-based system

- Rule:

  [Condition] [Action] [Evaluation]

- Rules compete, more than one can fulfil condition

- Evaluation criteria used to decide which to take, updated in each cycle
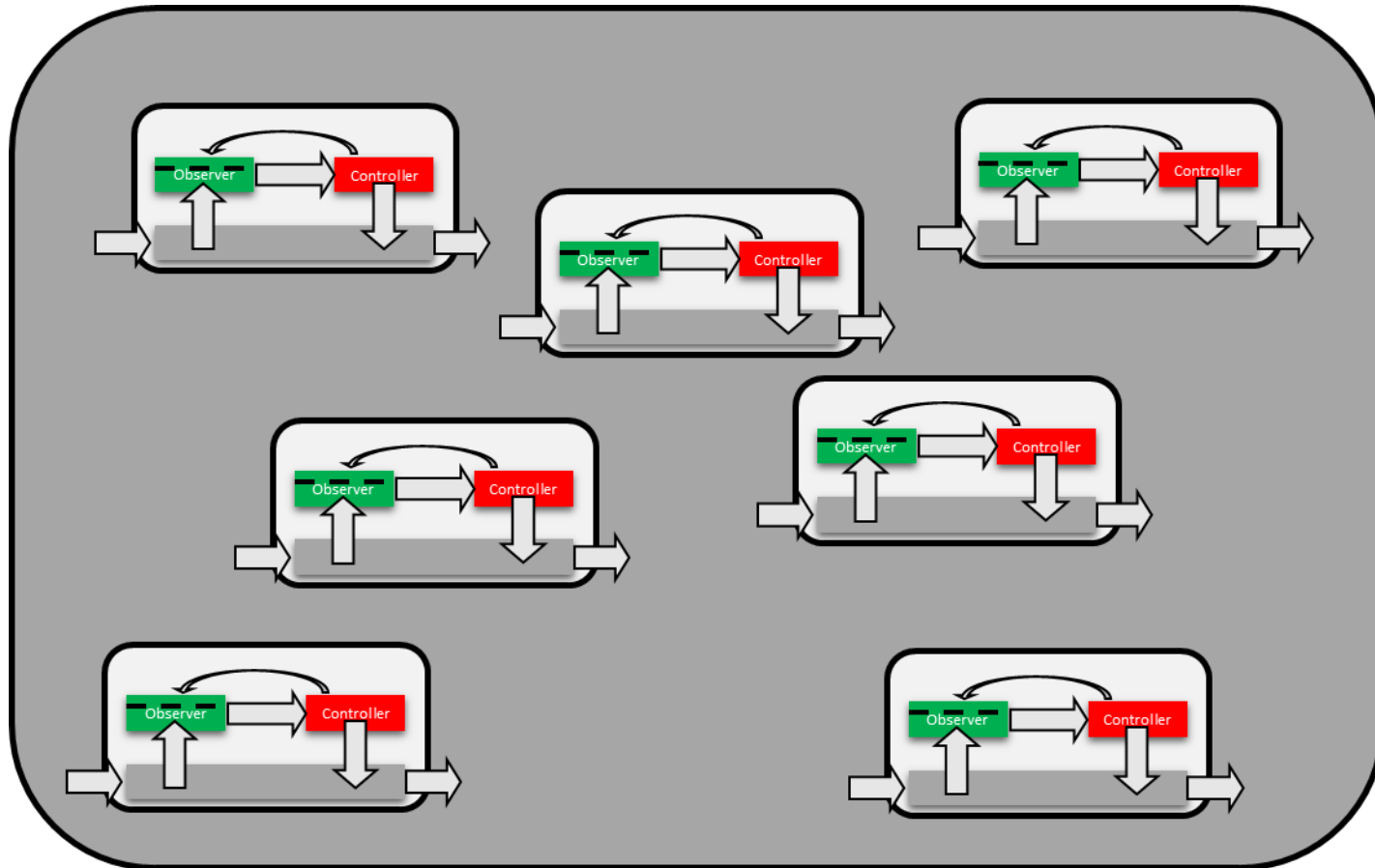
Traffic example

- Condition: vector with intervals for traffic flows per hour per turning

- Action: Green durations per traffic light
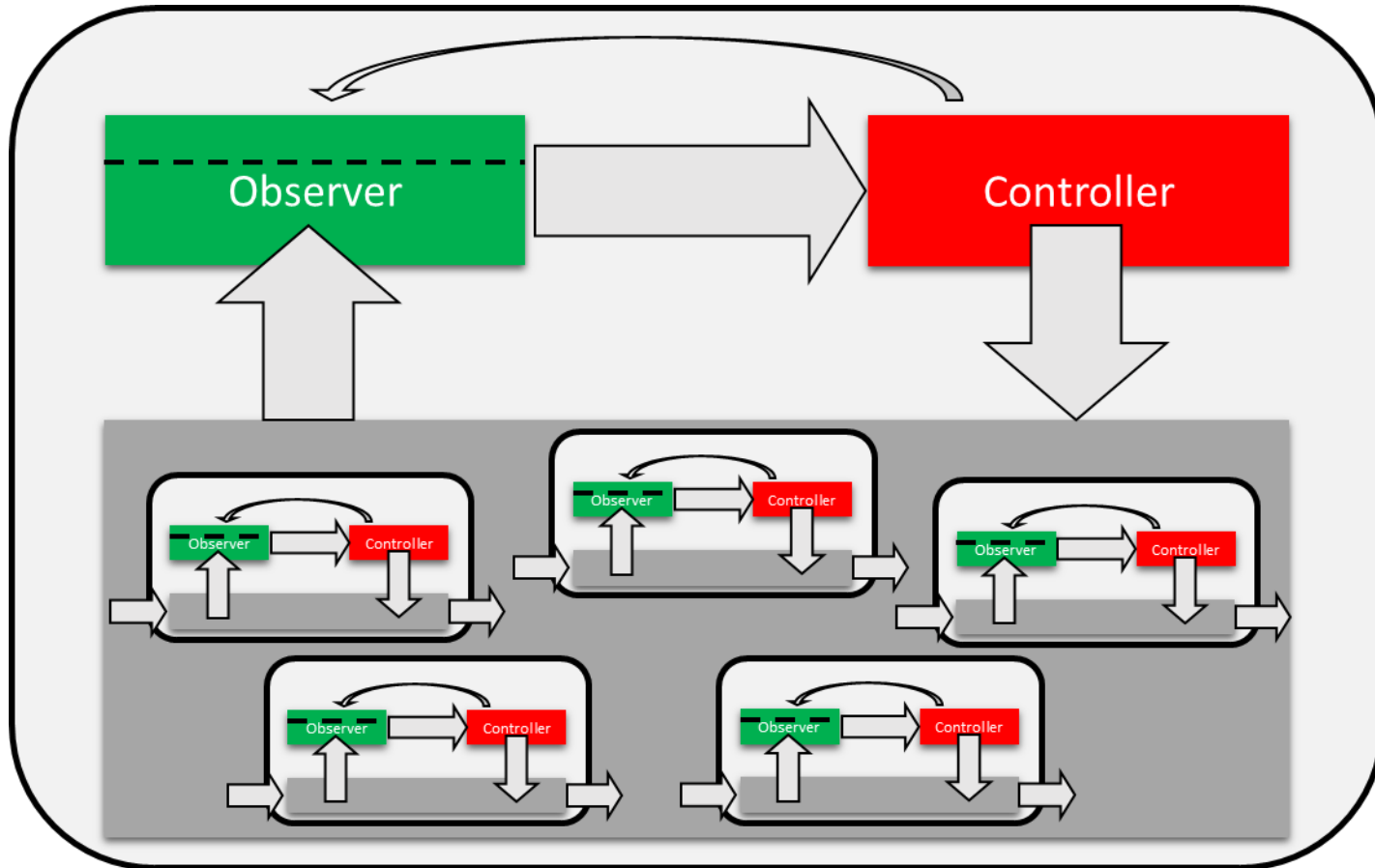
- Evaluation: Averaged delays occurring at intersection

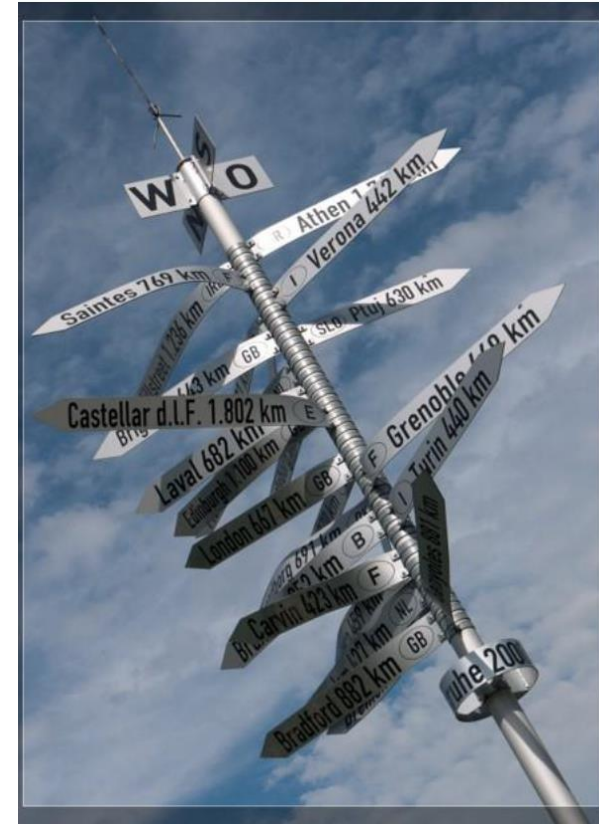# *Distribution variants*

Variant A: Fully centralised

Variant B: Fully decentralised
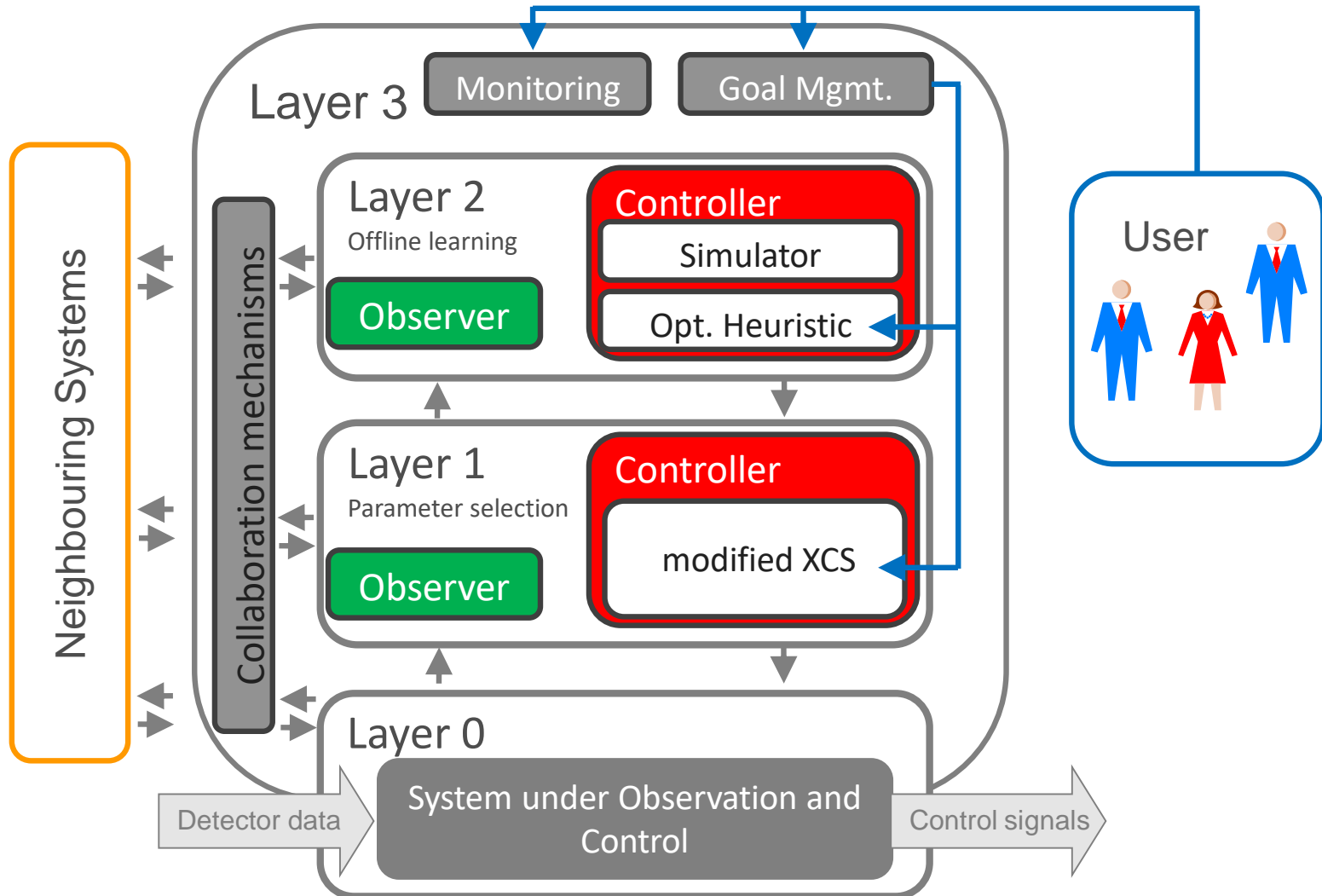
Variant C: Hybrid

Agenda

- General design concept for organic systems

- Observer/Controller architecture

- **Multi-level Observer/Controller framework**

- Conclusion and further readings

Multi-level observer/controller framework

Multi-level observer/controller framework (2)

- **Layer 0: Productive system**

  – Encapsulation of the SuOC

  – Provide access to observation and control interfaces

- **Layer 1: Parameter selection and online learning**

  – Observation of Layer 0:

    • Monitoring, pre-processing, data analysis, prediction, aggregation

    • Generation of situation description

  – Control:

    • Change parameters, structure, techniques to current conditions.

    • Select from existing rule-set, update [Evaluation] based on feedback.

- **Layer 2: Offline learning**

  – Observation of Layer 1: Missing or inappropriate knowledge

  – Control: Generate new rule, add to rule-based of Layer 1

- **Layer 3: Collaboration**

  – User: Self-explanation, goal modification

  – Other systems: Negotiation, communication, trust, …

Observer component at Layer 1

ORGANIC
COMPUTING

Components:

Observer
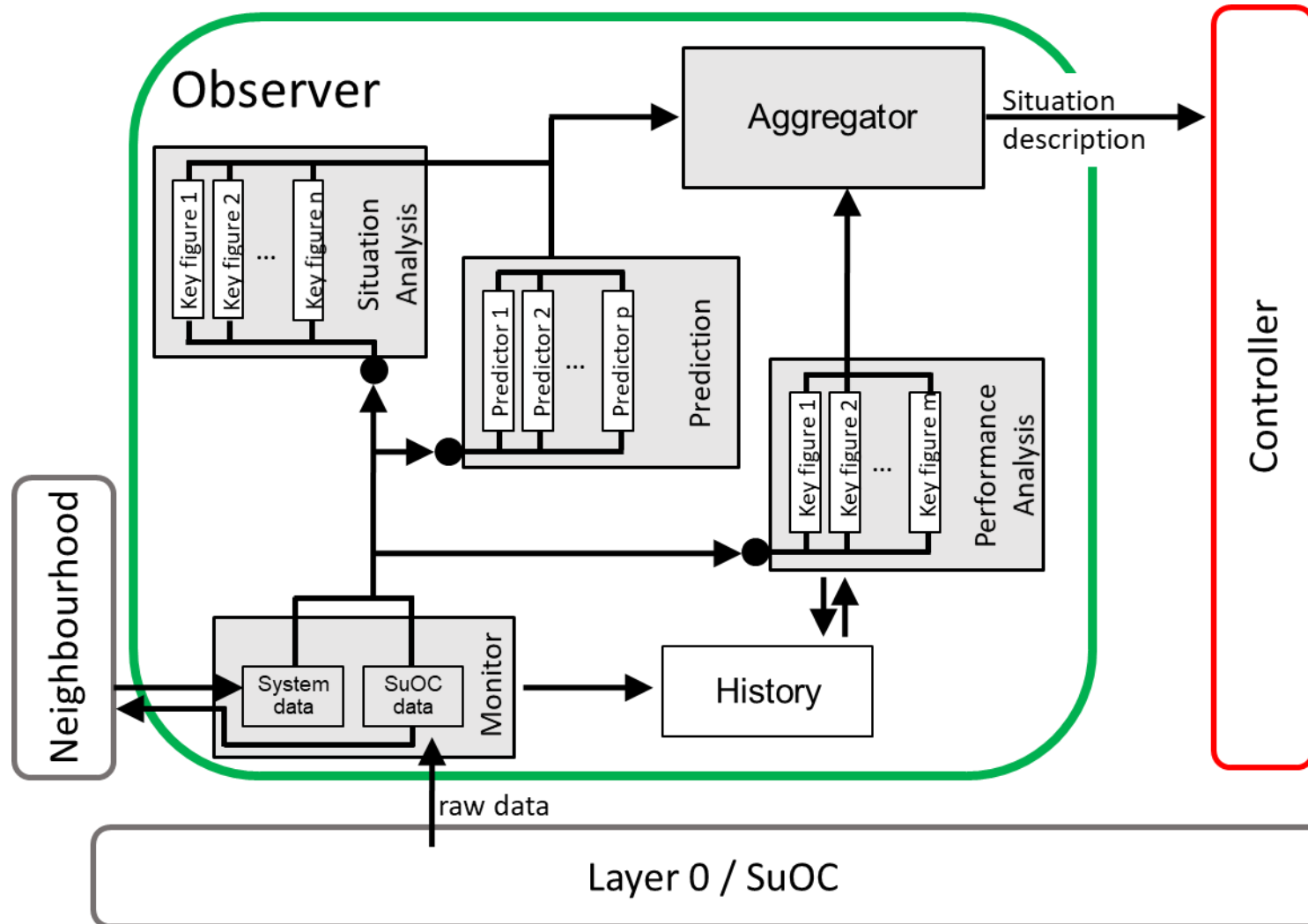
- Monitor: Gather sensor and status data

- Pre-processing: Estimate missing values, delete wrong values, smoothing, etc.

- Prediction: Forecasts for some of the underlying values

- Situation analysis: Find patterns, detect emergence, etc.

- Performance analysis: Extract values related to goals

- Aggregator: Build situation description

- History: List of observed data (for debugging and explanation)

Controller component at Layer 1 (2)
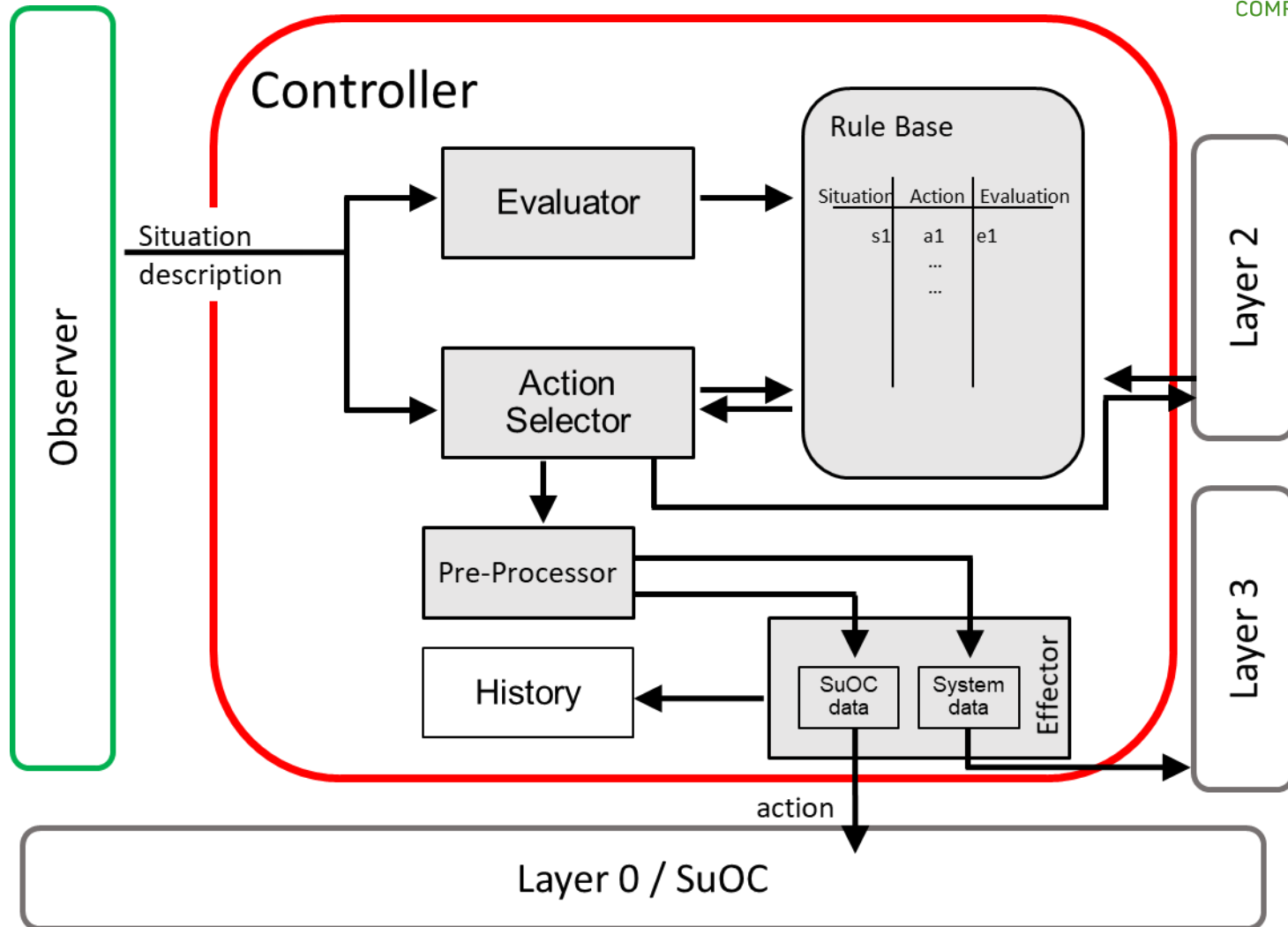
Components:

- Rule base: Set of rules, knowledge of the system

- Action selection: Select rule, action is encoded

- Effector: Activate action via effectors

- Evaluator: Determine success of last action(s), update rules

- Action history: List of situation-action mapping that have been performed

Controller

Agenda

- General design concept for organic systems

- Observer/Controller architecture

- Multi-level Observer/Controller framework

- Conclusion and further readings

Conclusion

This chapter:

- Introduced the basic design ideas used in organic systems.
- Refined these concepts by explaining the Observer/Controller approach.

By now, students should be able to:

- Outline the basic design concept used in OC.
- Sketch the Observer/Controller approach and the multi-level extension.
- Compare distribution variants.
- Explain the responsibilities of the contained components.

References

- Christian Müller-Schloer, Hartmut Schmeck, Theo Ungerer: „Organic Computing - A Paradigm Shift for Complex Systems". Springer 2011

- Sven Tomforde: "Runtime adaptation of technical systems". Südwest-deutscher Verlag für Hochschulschriften, ISBN-13: 978-3838131337, 2012.

Questions …?