TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik, Institut für Software- und Multimediatechnik, Lehrstuhl Softwaretechnologie

Hauptseminar "Autonomic Computing" 1. Seminar Day

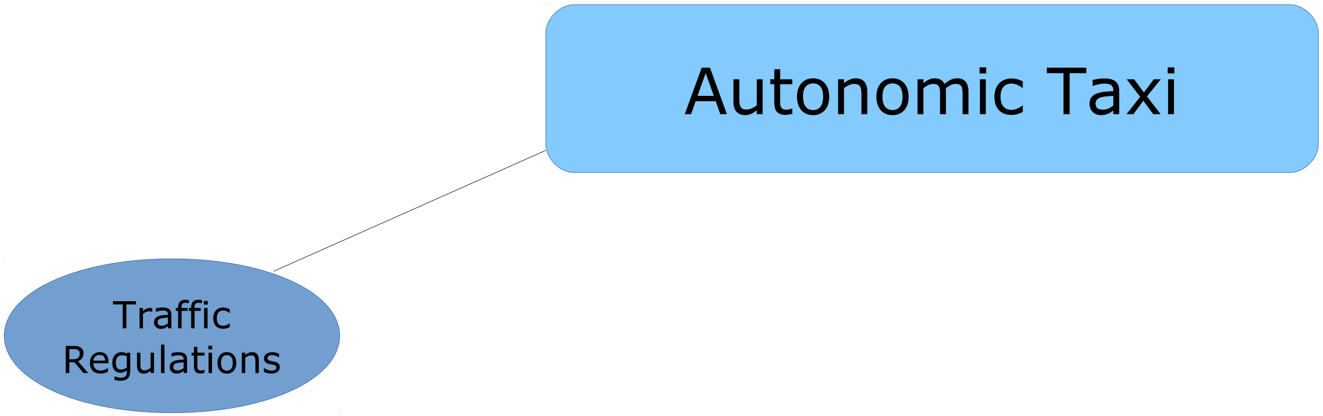
Dresden, 08.06.2016

Anja Reusch

Introduction

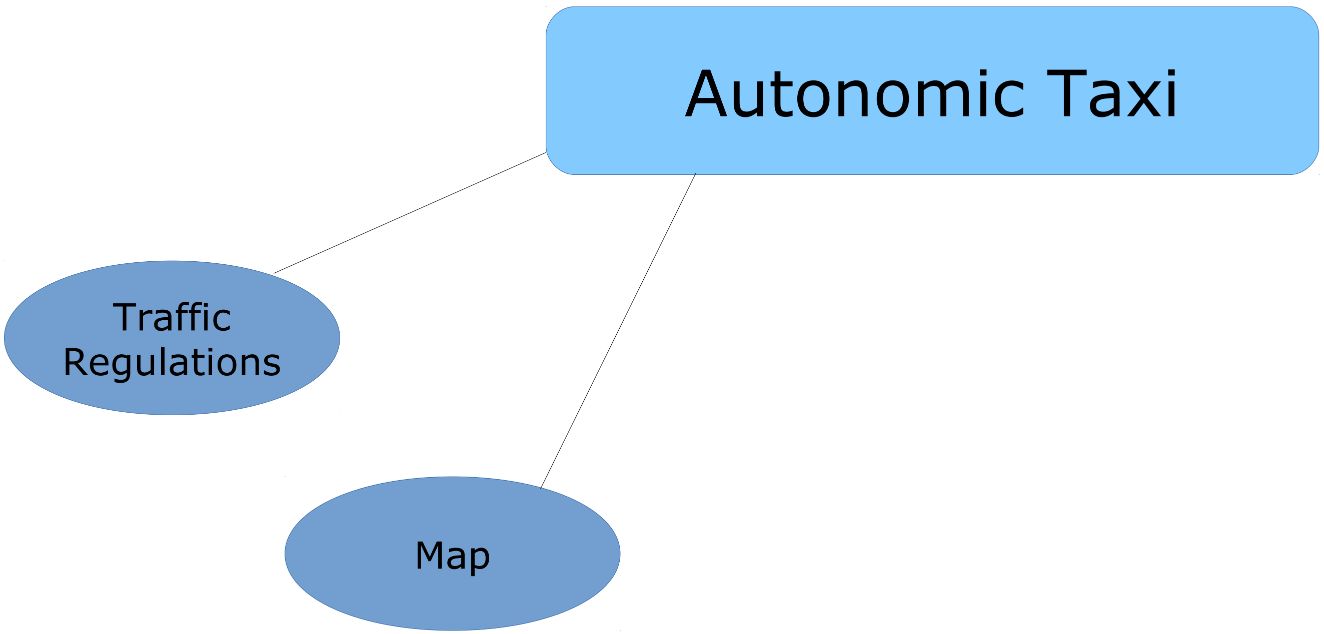
Autonomic Taxi

Introduction

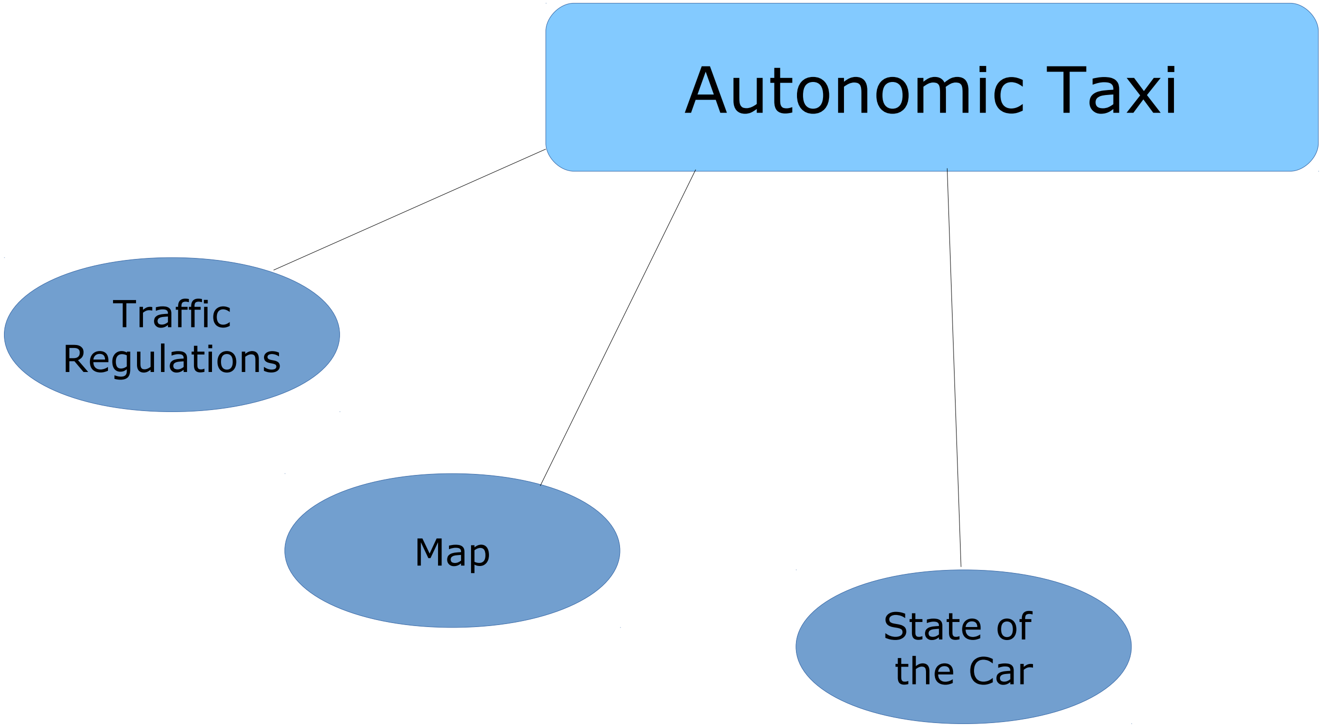


TECHNISCHE UNIVERSITÄT DRESDEN

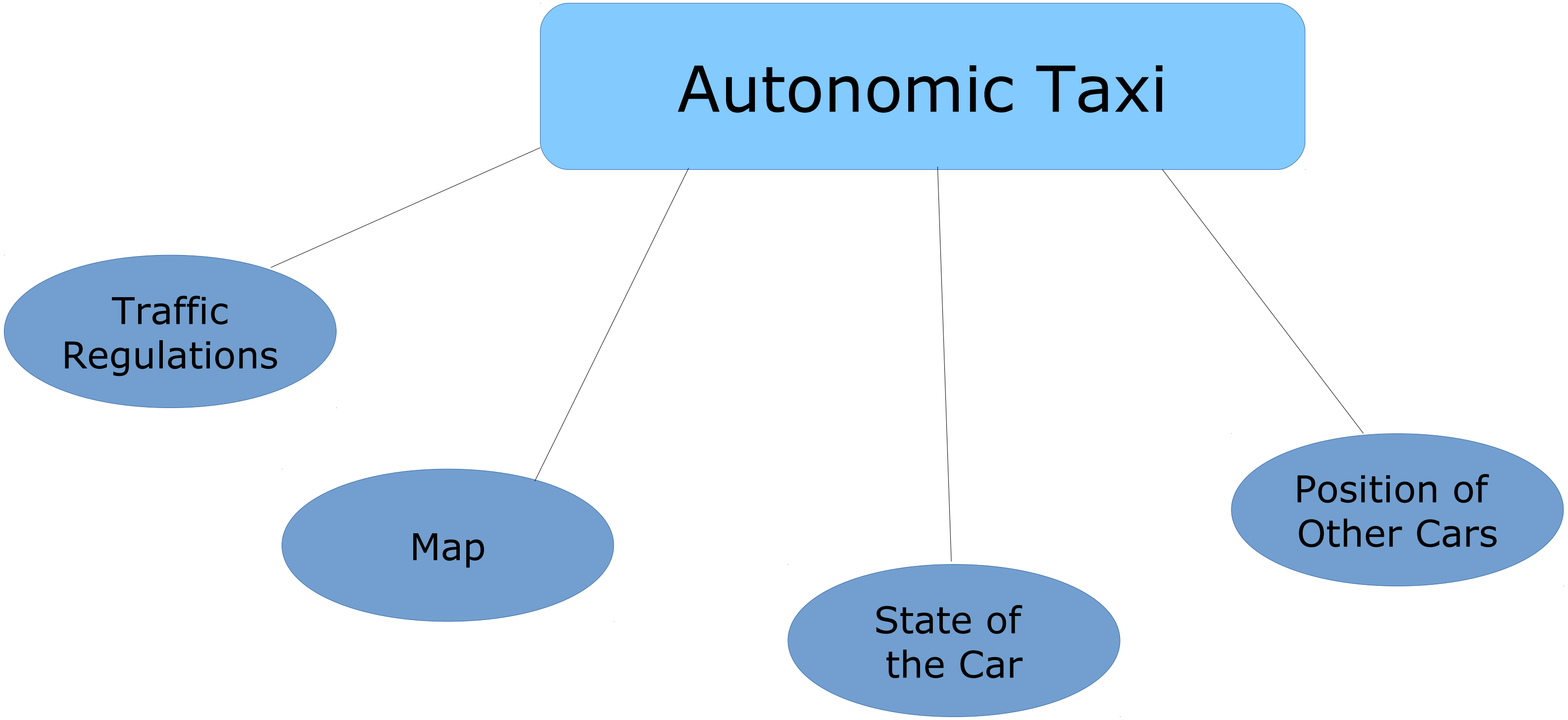
Introduction



Introduction



Introduction



Introduction

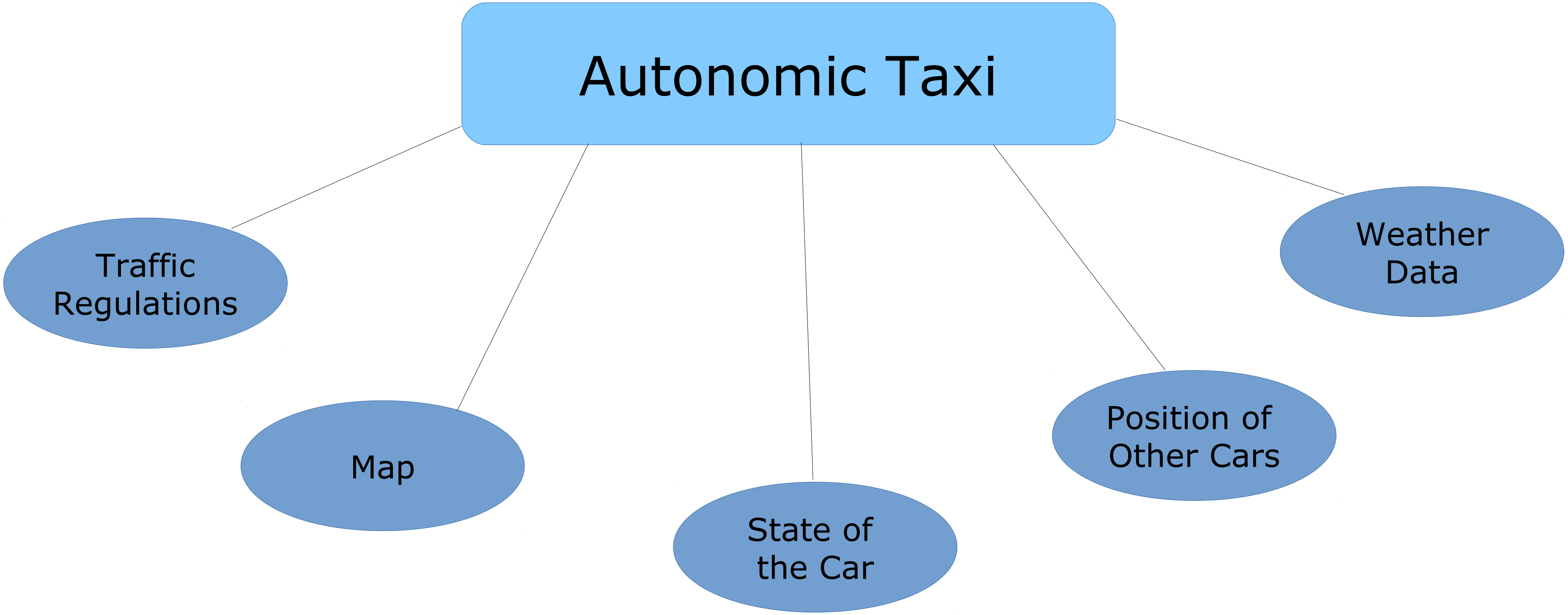


Table of Contents

Introduction

Example:
Autonomic Taxi

- Goals:
    - Carries passengers from one place to another
        - Route planning
    - Drives safely e.g.
        - Brakes if the car in front brakes,
        - Does not cross red traffic lights
        - Stays on the road

source: http://www.clipartlord.com/wp-content/uploads/2012/10/taxi-cab.png

Introduction

Example:
Autonomic Taxi

- Goals:
  - Carries passengers from one place to another
    - Route planning
  - Drives safely e.g.
    - Brakes if the car in front brakes,
    - Does not cross red traffic lights
    - Stays on the road
- Environment:
  - City
  - Roads
  - Highway
  - Traffic Lights
  - Other Cars
- Taxi gets tip after trip

source: http://www.clipartlord.com/wp-content/uploads/2012/10/taxi-cab.png

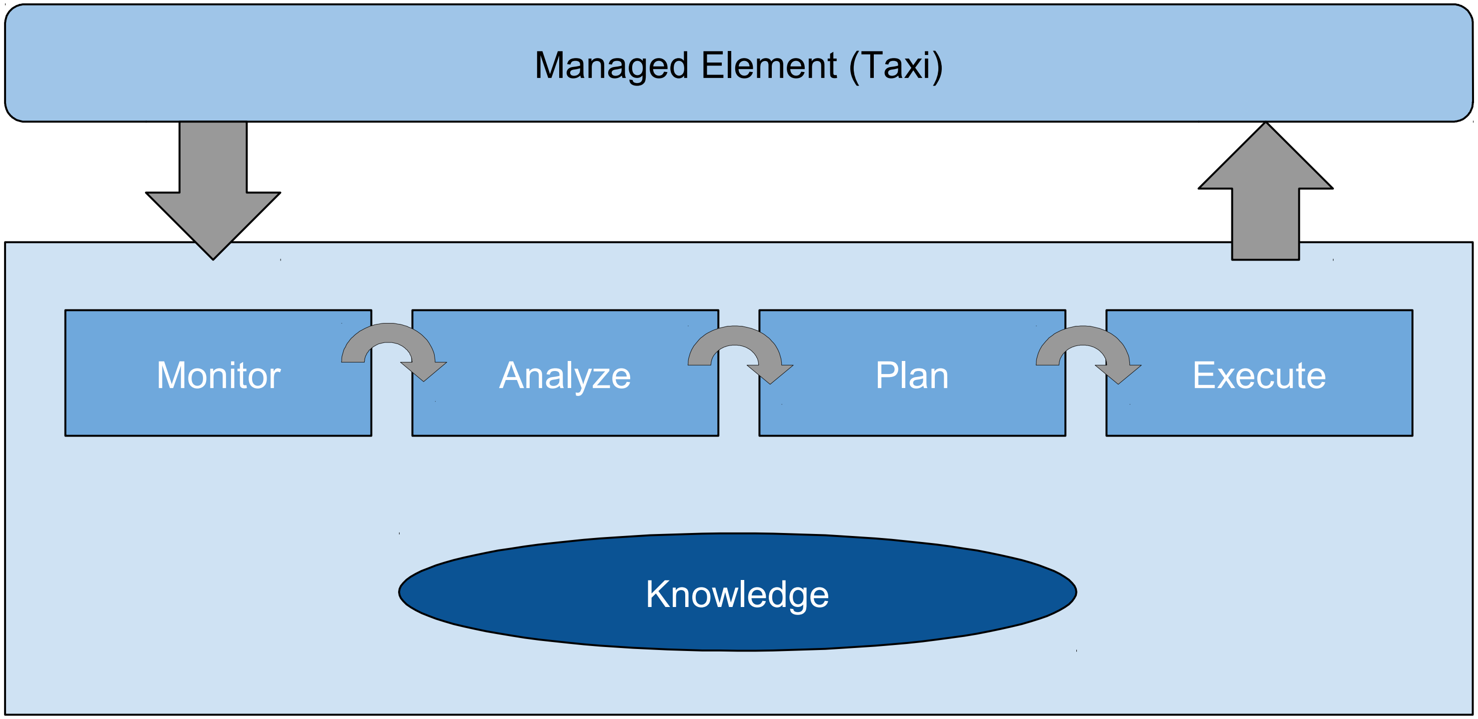
Background

Background

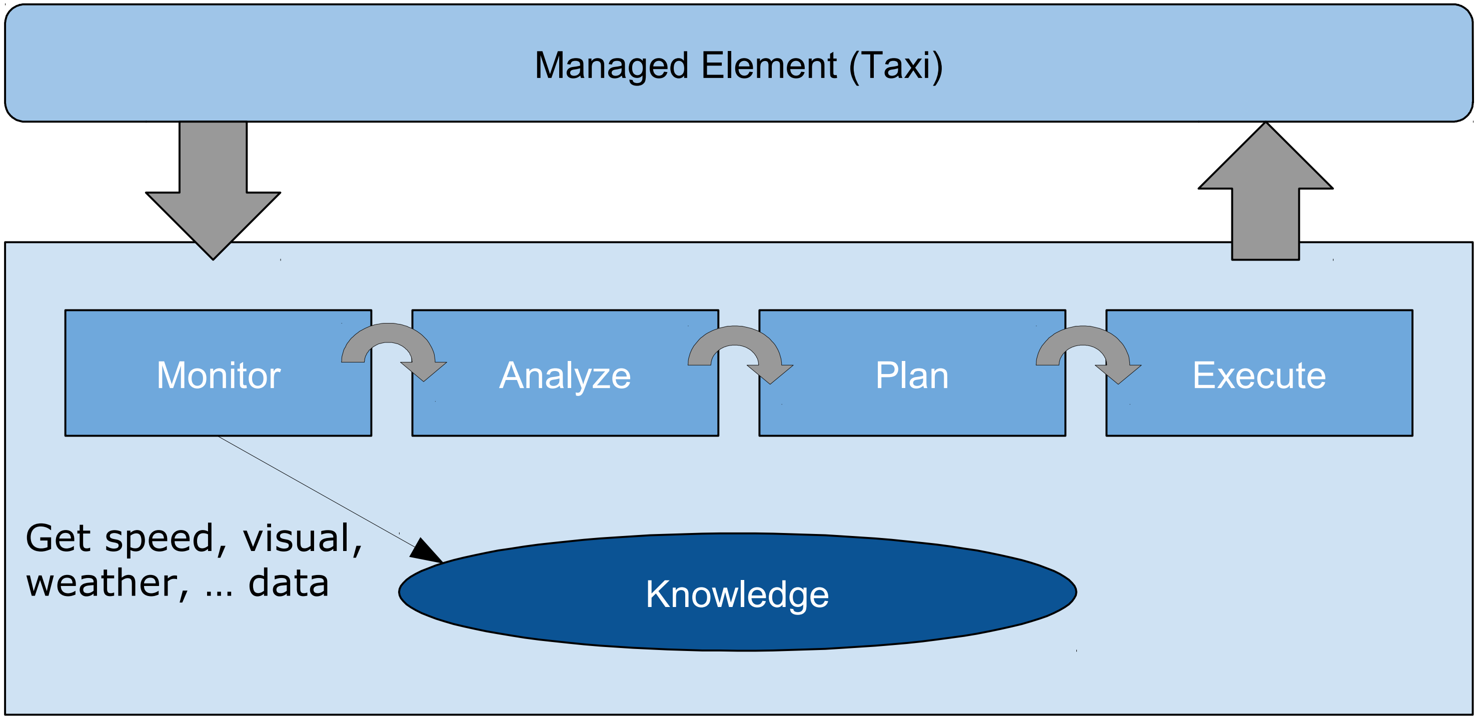
MAPE-K Architecture

- Introduced by IBM in 2005
- Used to structure automatic software systems
- Acronym for
    - Monitor
    - Analyze
    - Plan
    - Execute
    - Knowledge

Background

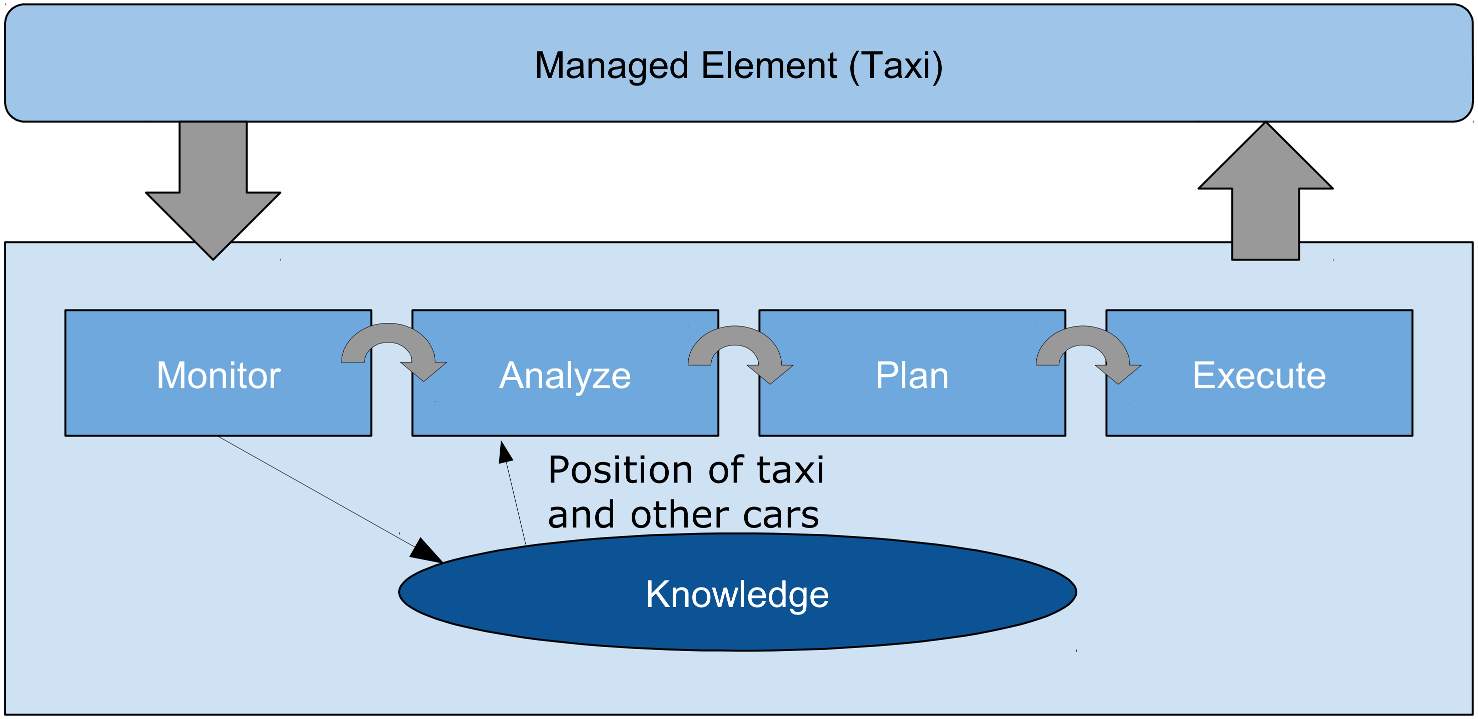

Managed Element (Taxi)
Monitor
Analyze
Plan
Execute
Knowledge

Background

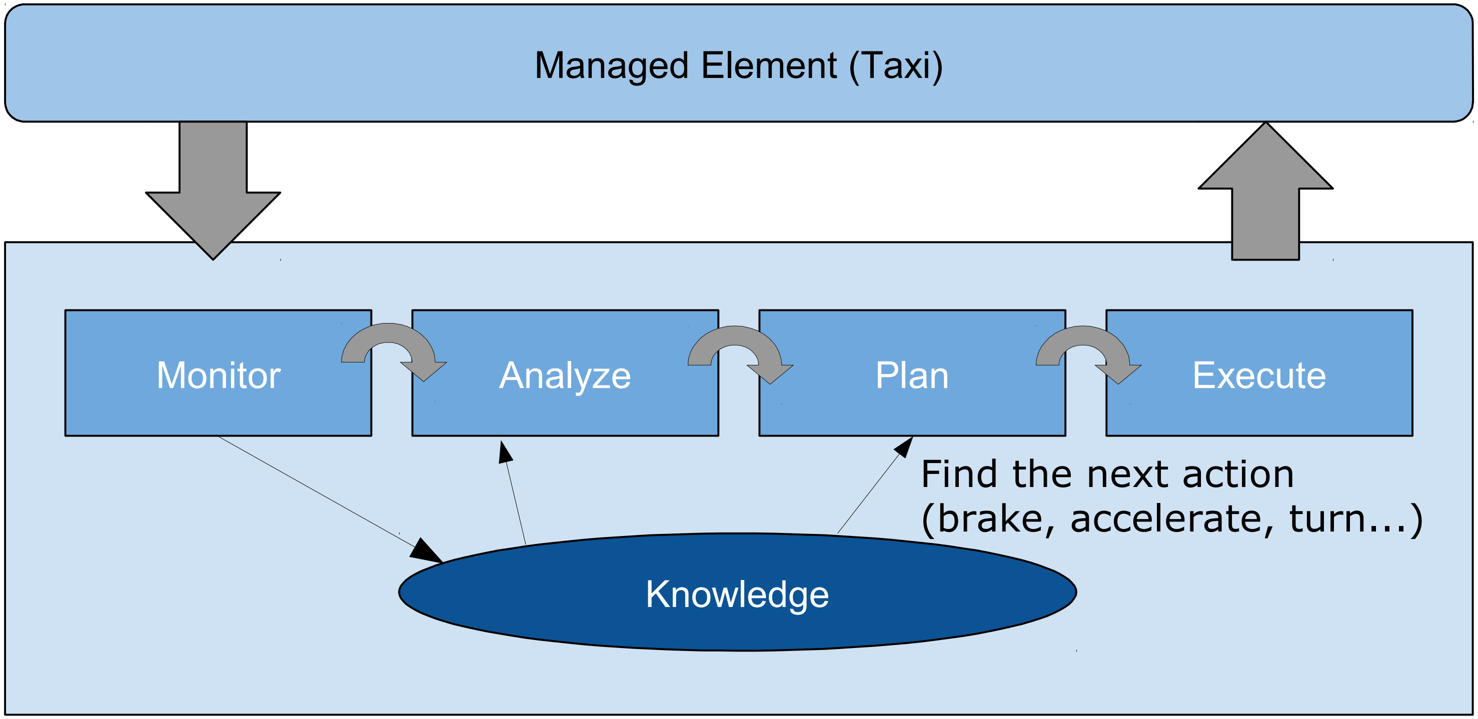


Managed Element (Taxi)
Monitor
Analyze
Plan
Execute
Get speed, visual,
weather, … data
Knowledge

Background


Managed Element (Taxi)
Monitor
Analyze
Plan
Execute
Position of taxi and other cars
Knowledge

Background



→ Knowledge is involved in the whole MAPE-K loop.

TECHNISCHE UNIVERSITÄT DRESDEN

Rule Based Autonomic Software System

Rule Based Autonomic Software System

- System is only based on rules
- Does not posses internal states or models of its environment
- Rules:
    - Action – Condition – Rules
    - Derived from system and business goals
    - Describe adaptation plans of the system

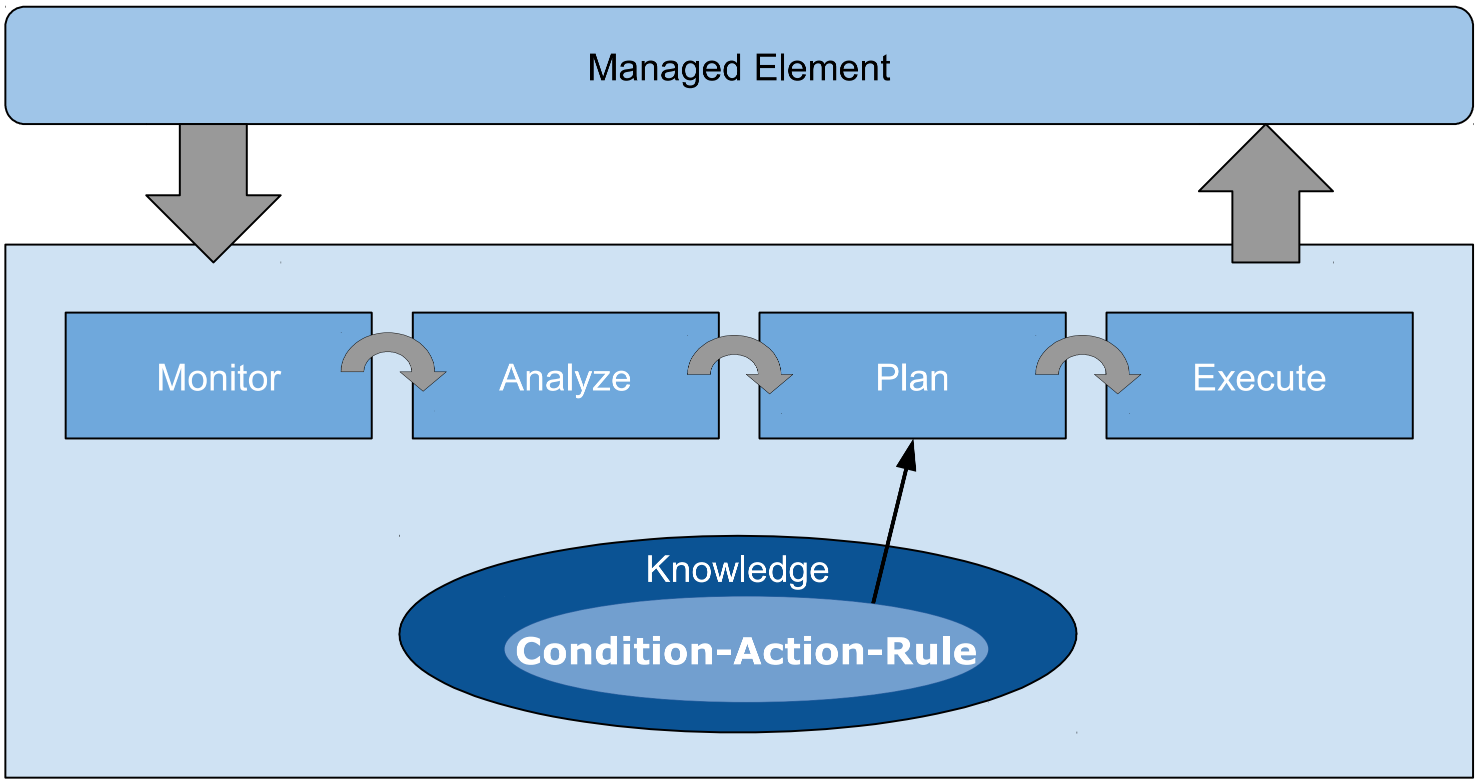
Rule Based Autonomic Software System

- System is only based on rules
- Does not posses internal states or models of its environment
- Rules:
    - Action – Condition – Rules
    - Derived from system and business goals
    - Describe adaptation plans of the system

- Example rule:

<policy> Smart Brake
<condition> Car in front brakes
<action> Brake
</policy>

Rule Based Autonomic Software System



TECHNISCHE UNIVERSITÄT DRESDEN

Rule Based ASS - Example

Taxi

Rule Based System

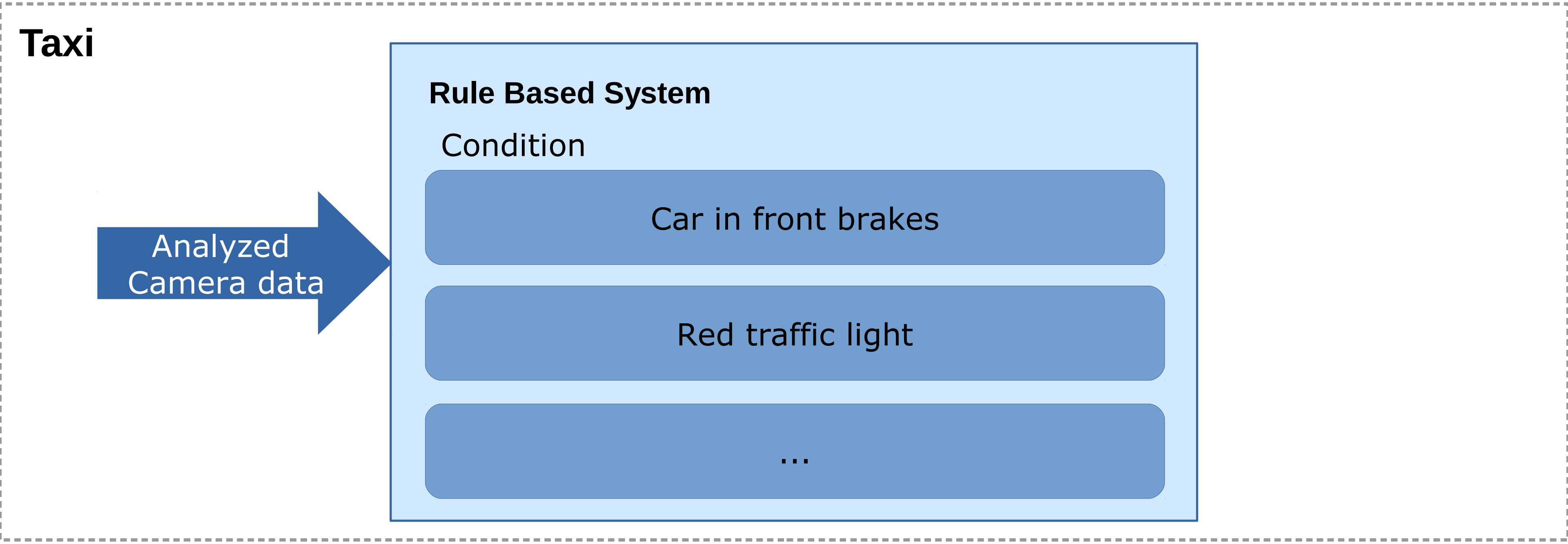
Condition

Car in front brakes

Red traffic light

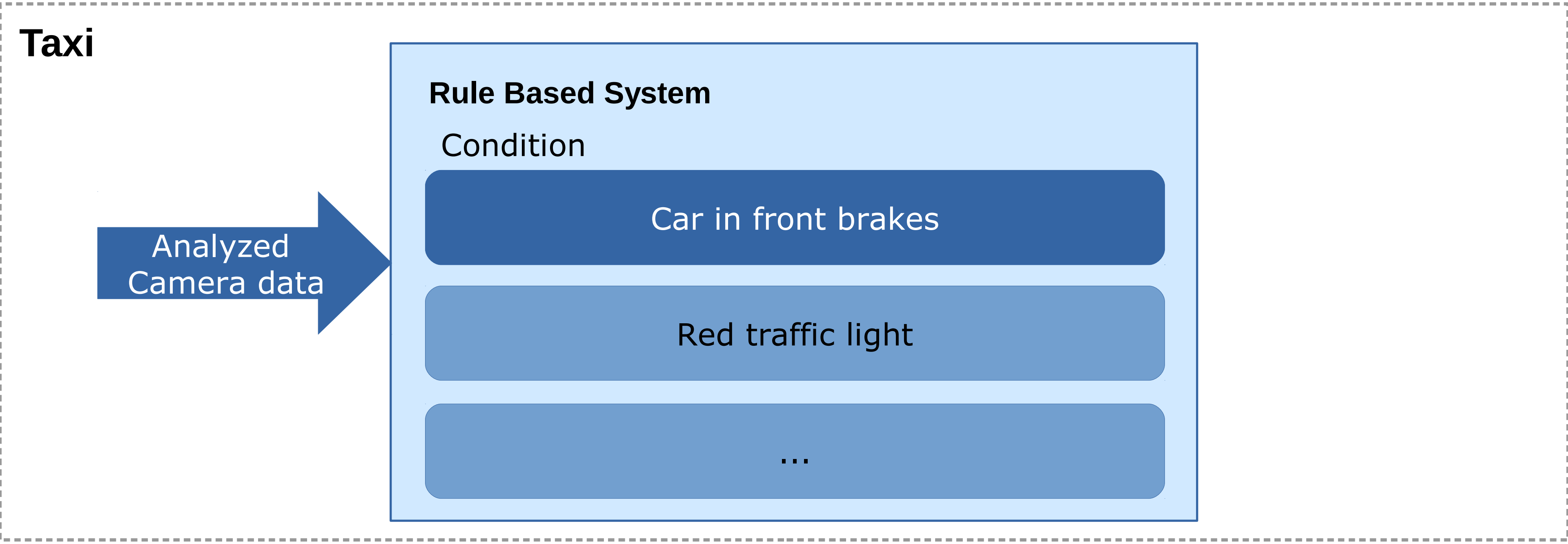
...

Rule Based ASS - Example

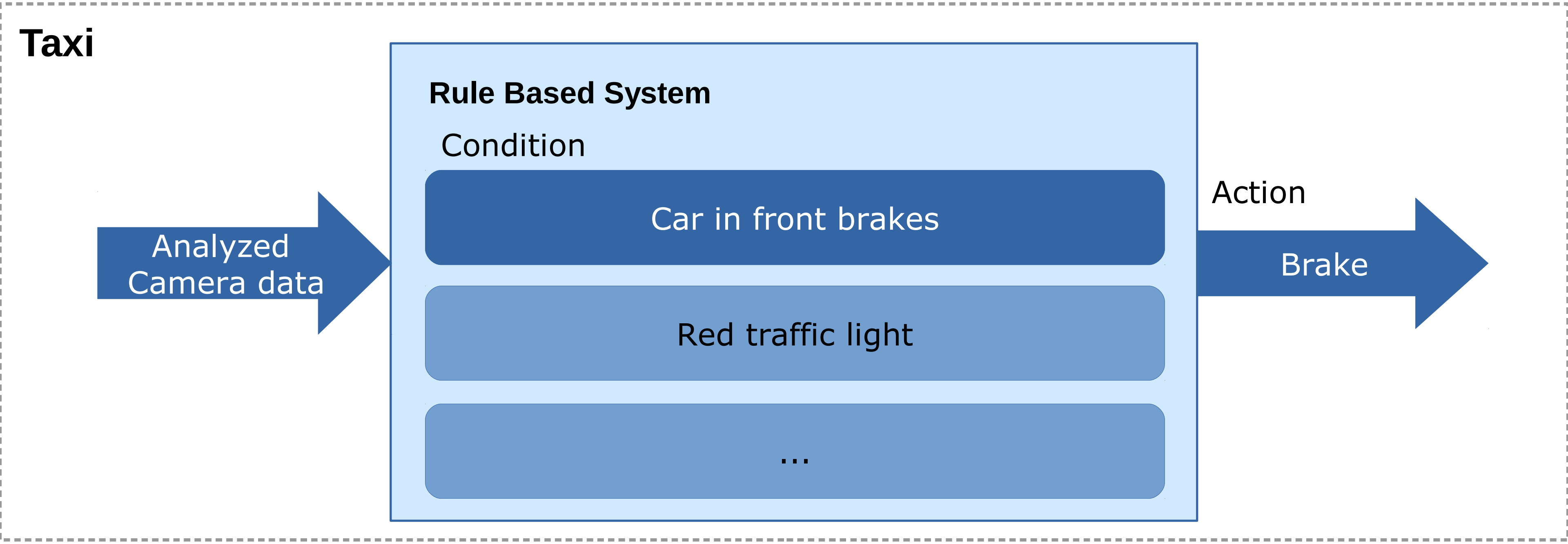


TECHNISCHE UNIVERSITÄT DRESDEN

Rule Based ASS - Example



Rule Based ASS - Example



Rule based ASS - Evaluation

Advantages	Disadvantages
Decisions are made quickly.	Rules can not be adapted to any kind of changes
Lightweight: no need of lot of memory or processing time	Conflicts between rules
	ASS does not consider the state or actions in the past.

TECHNISCHE UNIVERSITÄT DRESDEN

Model Based Autonomic Software System

Model Based Autonomic Software System

- System uses models to represent components and relations in the world
- Model
  - Contains information on the state of the managed element
  - Updated through e.g. fresh sensor readings
  - Represented as graphs

Model Based Autonomic Software System

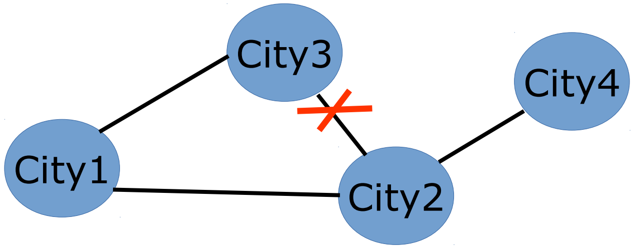
- System uses models to represent components and relations in the world
- Model
  - Contains information on the state of the managed element
  - Updated through e.g. fresh sensor readings
  - Represented as graphs
- Example Models
  - `function velocity(time) = current_acceleration * time`

TECHNISCHE UNIVERSITÄT DRESDEN

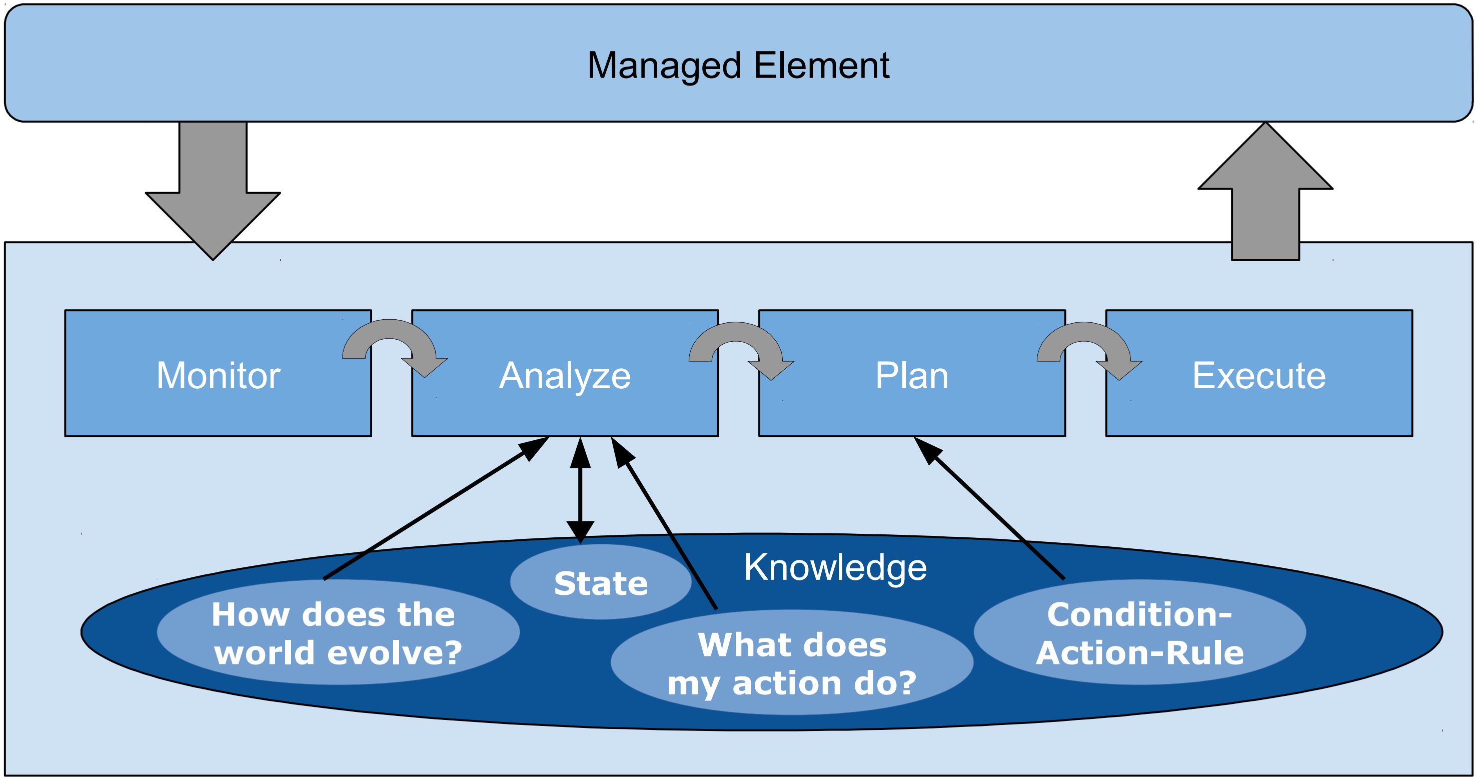
Model Based Autonomic Software System

- System uses models to represent components and relations in the world
- Model
  - Contains information on the state of the managed element
  - Updated through e.g. fresh sensor readings
  - Represented as graphs

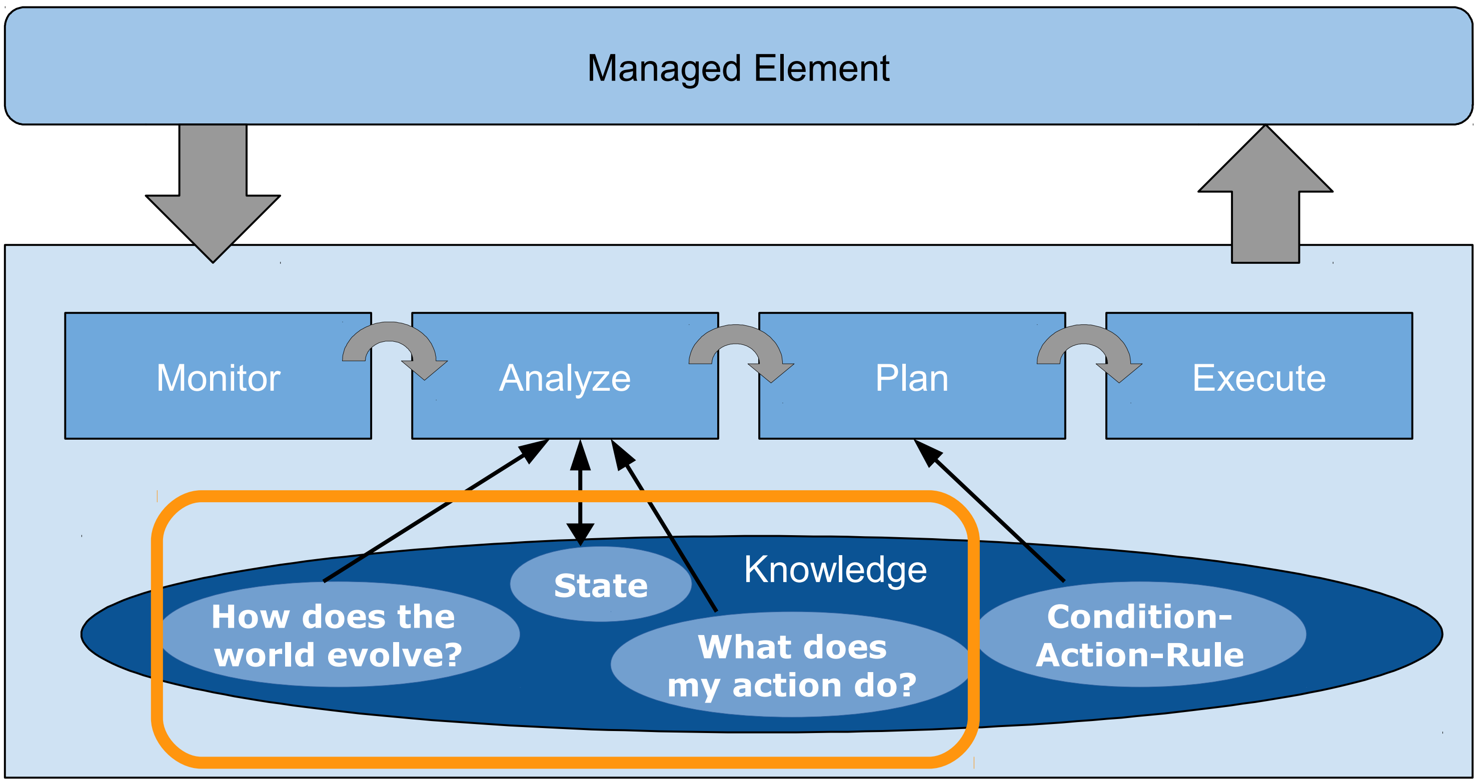
- Example Models
  - `function velocity(time) = current_acceleration * time`
  - Map



Model Based Autonomic Software System



Model Based Autonomic Software System



Model Based ASS

Taxi
State: {
Velocity: 90 km/h,
Position: …,
Passengers: …,
…},
Model: {
Map,
Velocity function
...},
Rules: {
Smart Brake,
Red Traffic Light,
...}
Last Action:
"switch to left lane"

Model Based ASS

Taxi

State: {
Velocity: 90 km/h,
Position: …,
Passengers: …,
…},

Model: {
Map,
Velocity function
...},

Rules: {
Smart Brake,
Red Traffic Light,
...}

Last Action:
"switch to left lane"

Action:
accelerate

Model Based ASS

Taxi

<u>State:</u> {
Velocity: 90 km/h,
Position: …,
Passengers: …,
…},
<u>Model:</u> {
Map,
Velocity function
…},
<u>Rules:</u> {
Smart Brake,
Red Traffic Light,
…}
<u>Last Action:</u>
"switch to left lane"

Action:
accelerate

Taxi

<u>State:</u> {
Velocity: **100 km/h**,
Position: …,
Passengers: …,
…},
<u>Model:</u> {
Map,
Velocity function
…},
<u>Rules:</u> {
Smart Brake,
Red Traffic Light,
…}
<u>Last Action:</u>
"accelerate"

TECHNISCHE UNIVERSITÄT DRESDEN

Model based ASS - Evaluation

Advantages	Disadvantages
ASS can make predictions about the future behavior.	Synchronize many models
ASS can avoid state flapping.	Hard to change goals

Reinforcement Learning

Reinforcement Learning

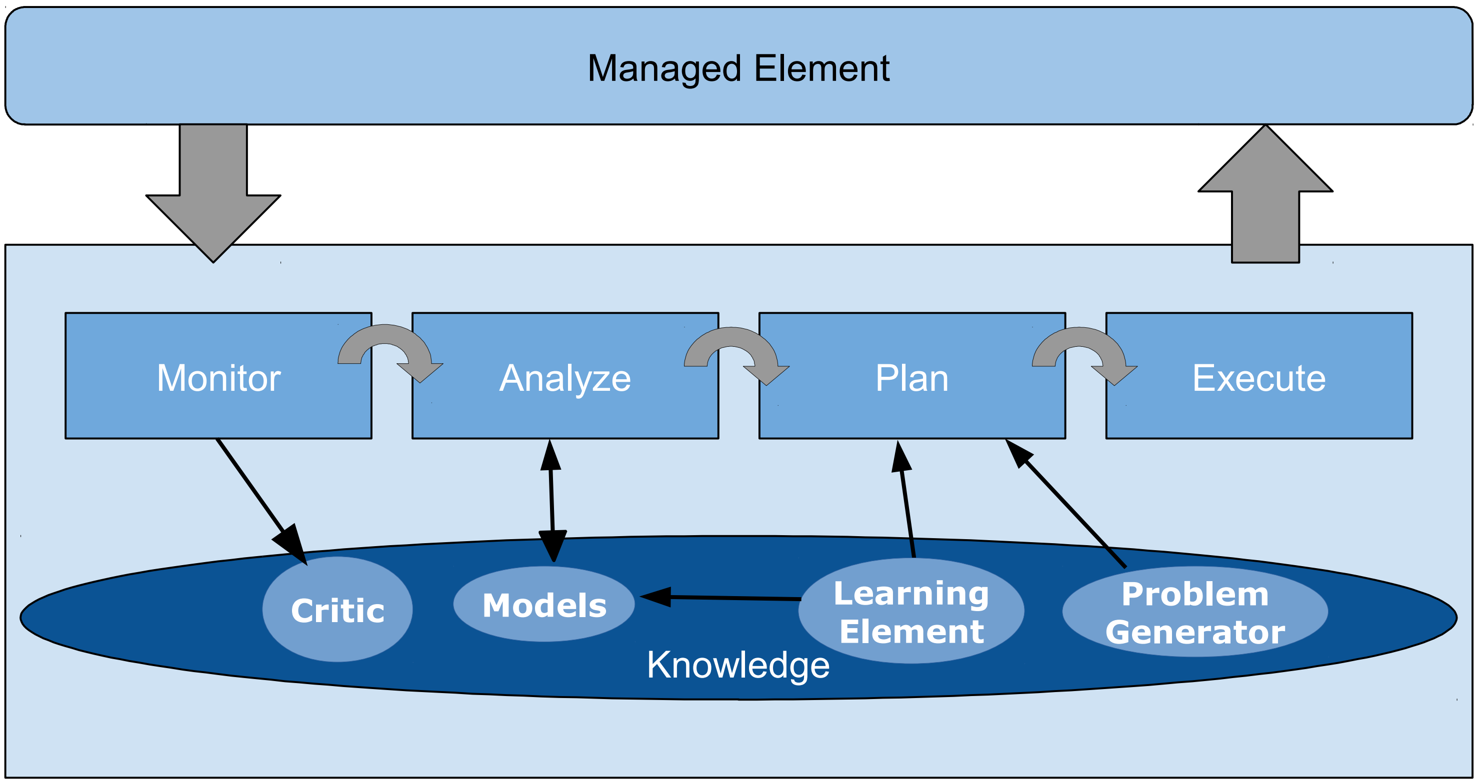
- Learns policies from performed actions
- Evaluation of usefulness of actions using rewards
- Modifies knowledge i.e. changes the models or utility functions
- Components:
    - Critic (with Reward)
    - Problem Generator
    - Learning Element
    - Performance Standard

Reinforcement Learning

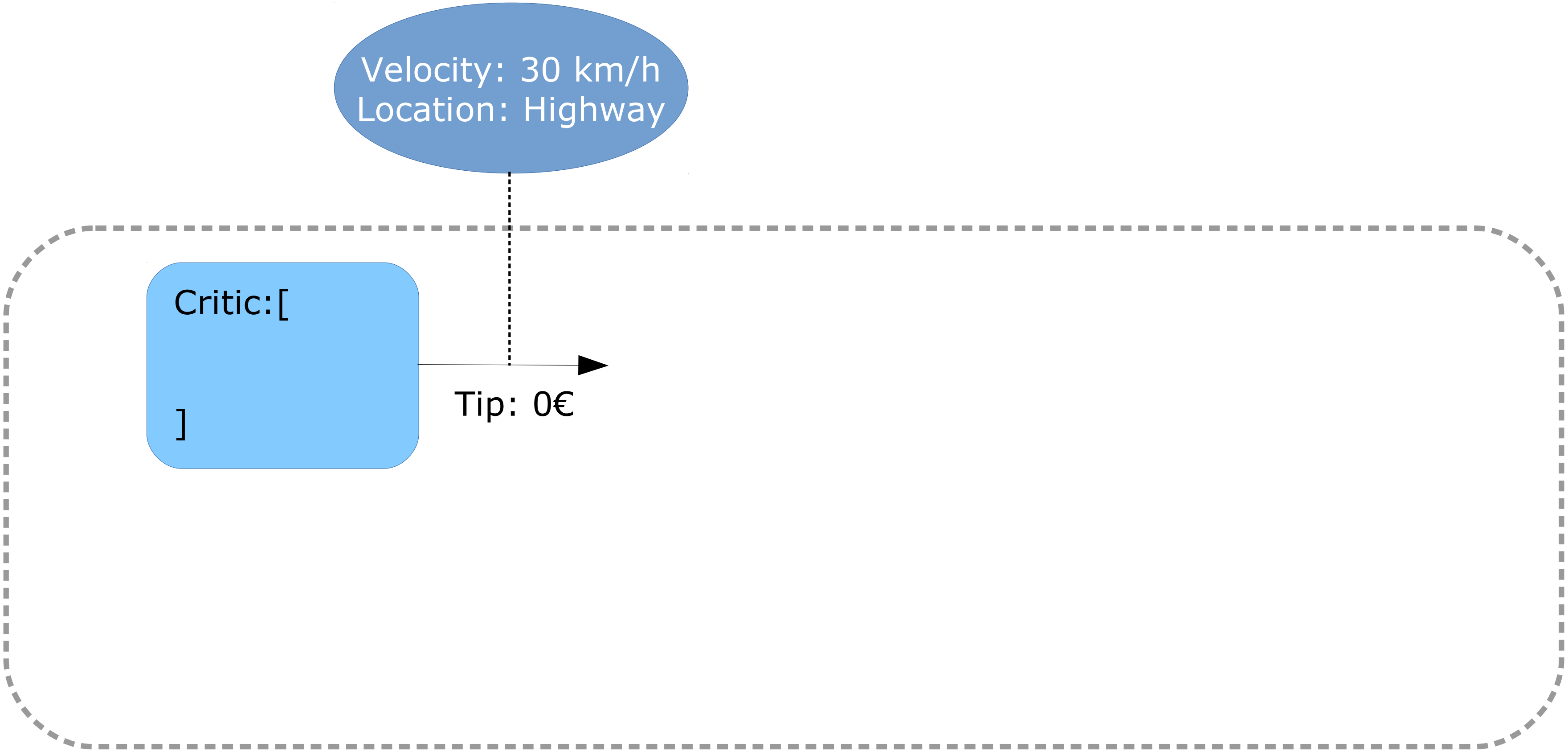
- Learns policies from performed actions
- Evaluation of usefulness of actions using rewards
- Modifies knowledge i.e. changes the models or utility functions
- Components:
    - Critic (with Reward)
    - Problem Generator
    - Learning Element
    - Performance Standard

- Example
    - Reward: tip from Passengers
    - Performance Standard: more tip is better

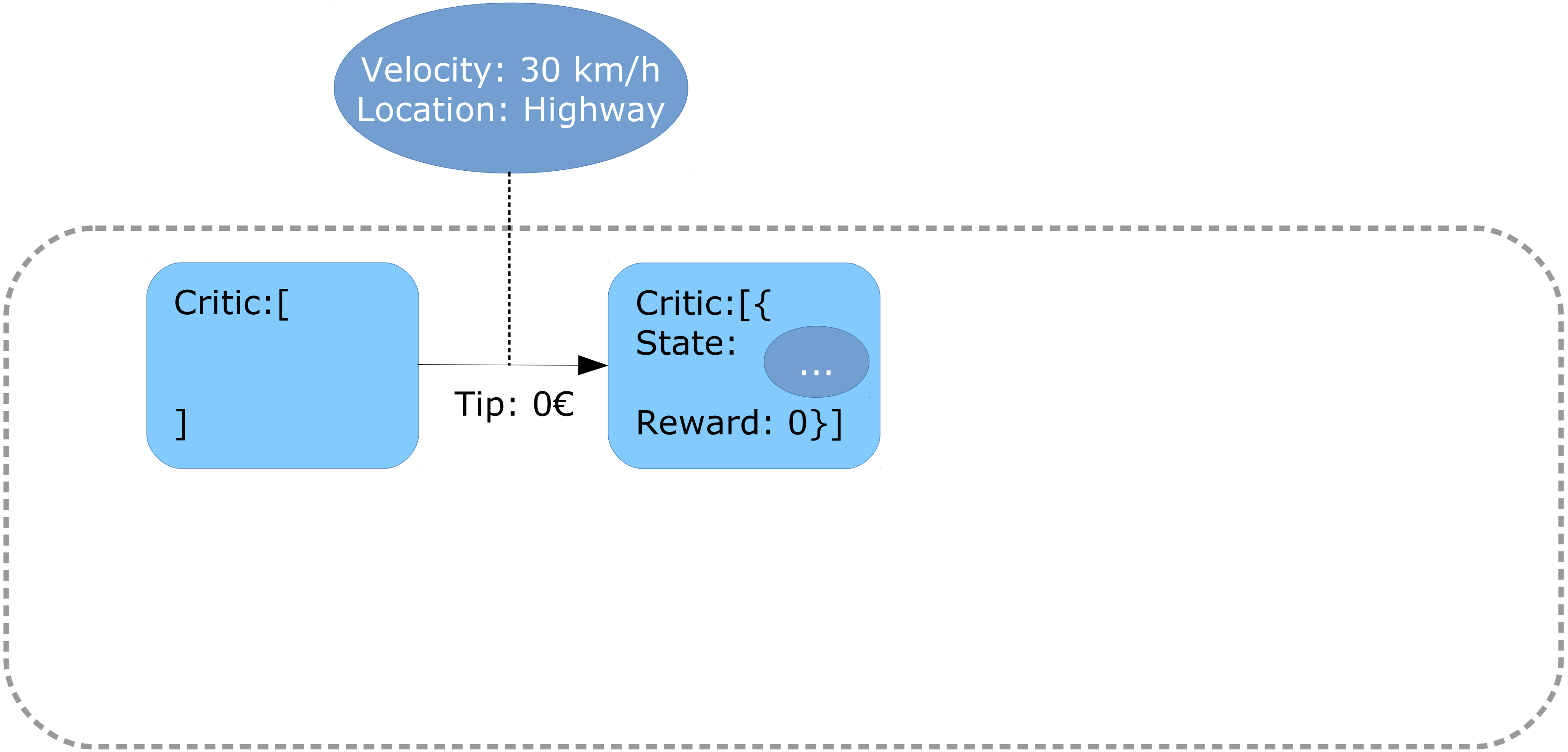
Reinforcement Learning



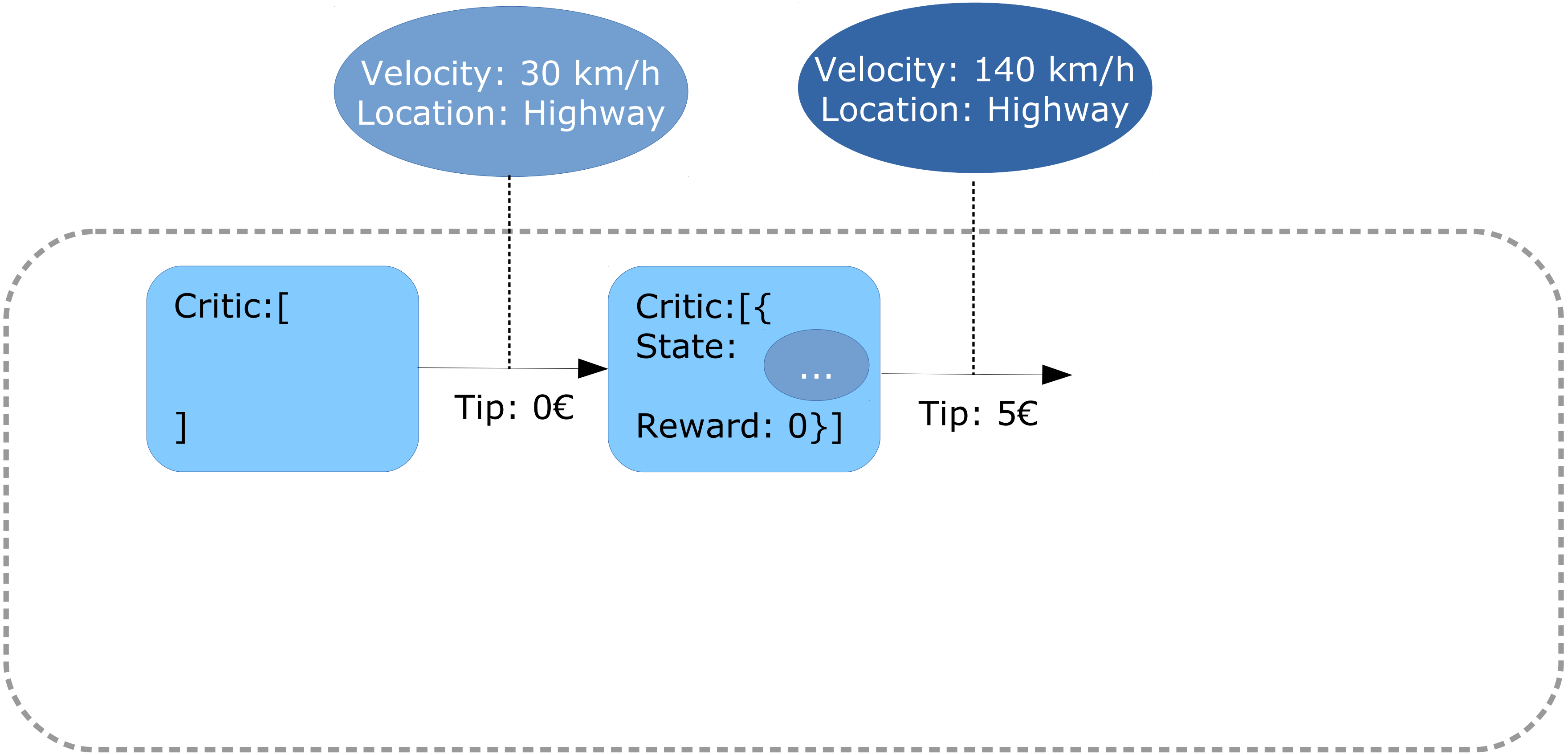
Reinforcement Learning - Example



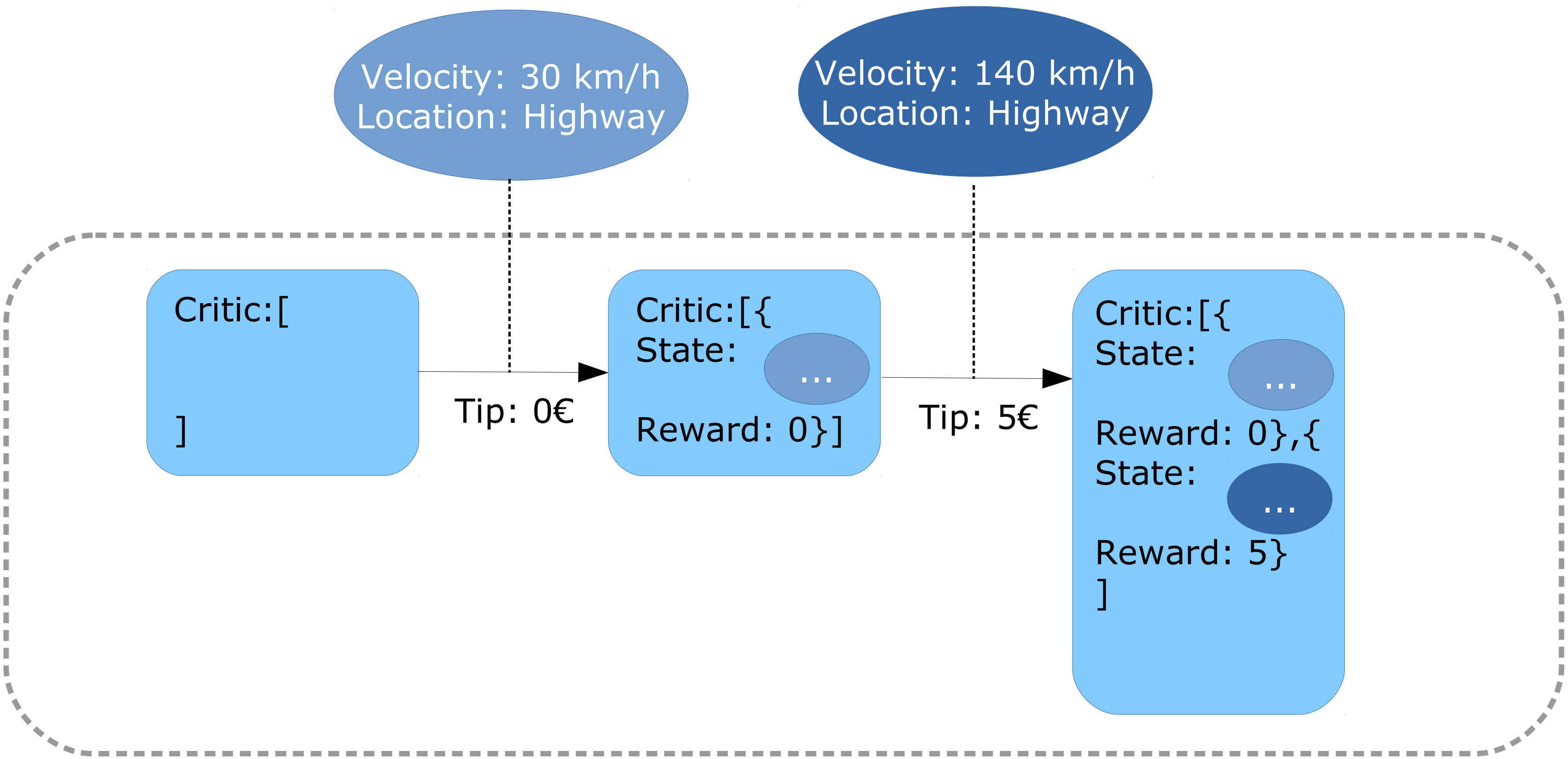
Reinforcement Learning - Example



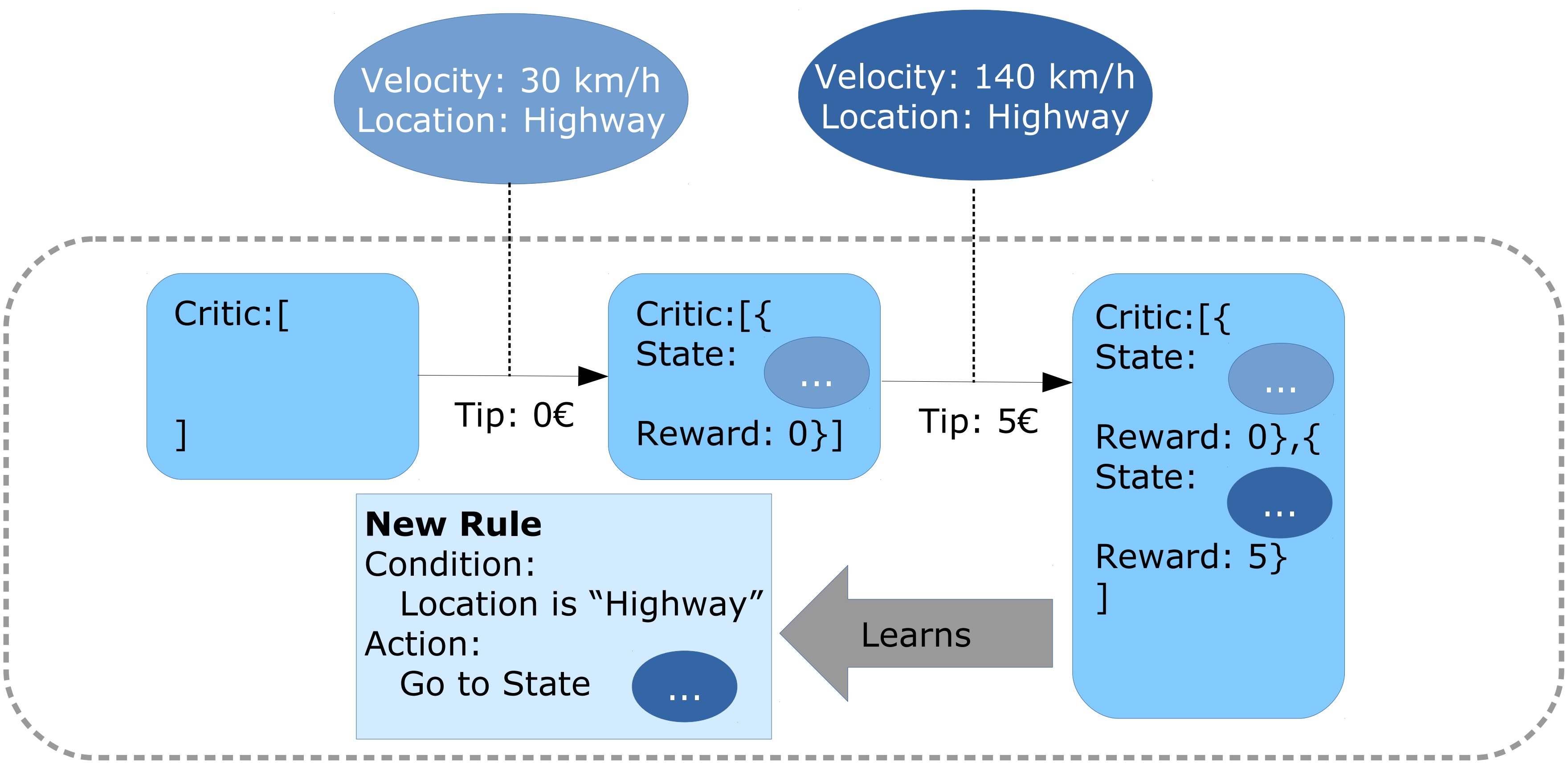
Reinforcement Learning - Example



Reinforcement Learning - Example



Reinforcement Learning - Example



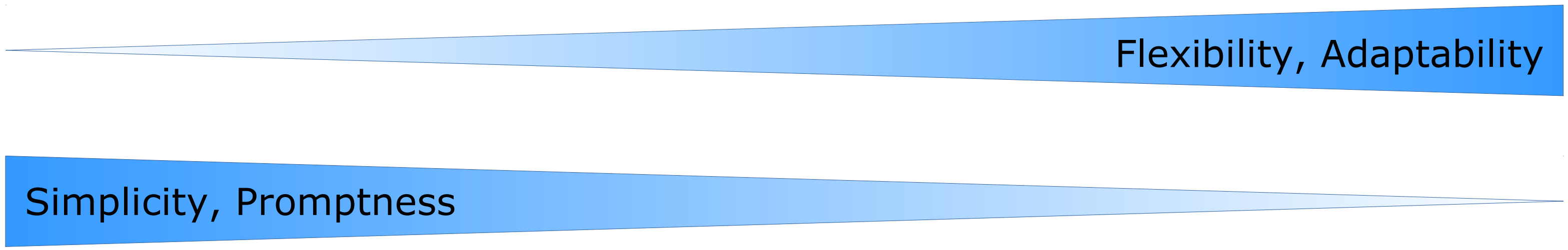
Reinforcement Learning - Evaluation

Advantages	Disadvantages
ASS can operate in an initially unknown environment.	Needs time for training the
ASS can become more competent than its initial knowledge	Needs more memory and processing time

Recommendation and Conclusion

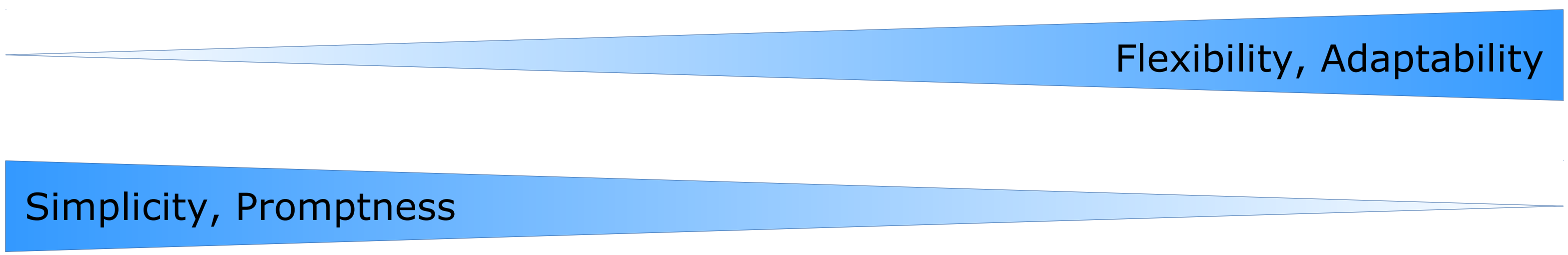
TECHNISCHE UNIVERSITÄT DRESDEN

Conclusion and Recommendations



Rule Based ASS **Model Based ASS** **Reinforcement Learning**

Conclusion and Recommendations



Rule Based ASS	Model Based ASS	Reinforcement Learning

- Rule Based ASS: High Performance System, Micro-controller

Conclusion and Recommendations

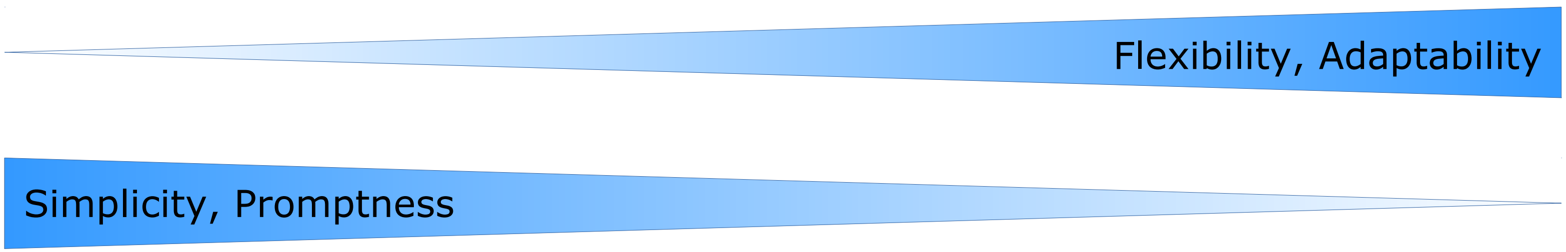
Flexibility, Adaptability

Simplicity, Promptness

Rule Based ASS | **Model Based ASS** | **Reinforcement Learning**

- Rule Based ASS: High Performance System, Micro-controller
- Model Based ASS: Industrial Robots

Conclusion and Recommendations



Rule Based ASS | **Model Based ASS** | **Reinforcement Learning**

- Rule Based ASS: High Performance System, Micro-controller
- Model Based ASS: Industrial Robots
- Reinforcement Learning: Autonomic Vehicle, artificial personal assistant

Thank you!

… Questions?

TECHNISCHE UNIVERSITÄT DRESDEN

References

Intelligent agents: Theory and practice

Wooldridge, Michael and Jennings, Nicholas R and others, Cambridge Univ Press, 1995

An architectural blueprint for autonomic computing. IBM White Paper (2006)

A semantic web primer, Antoniou, G., Van Harmelen, F., MIT press (2004)

The vision of autonomic computing,

Kephart, J.O., Chess, D.M., Computer 36(1), 41–50 (2003)

An artificial intelligence perspective on autonomic computing policies,

Kephart, J.O., Walsh, W.E., In:Policies for Distributed Systems and Networks, 2004.POLICY2004.Proceedings. Fifth IEEE International Workshop on. pp. 3–12. IEEE (2004)

Autonomic Computing: Principles, design and implementation,

Lalanda, P., McCann, J.A., Diaconescu, A., Springer Science & Business Media (2013)

Knowledge representation and reasoning,

Levesque, H.J., Annual review of computer science 1(1), 255–287 (1986)

Reinforcement learning in autonomic computing,

Tesauro, G., A manifesto and case studies. Internet Computing, IEEE 11(1), 22–30 (2007)