Online Learning for Self-Healing and Self-Optimization

Project Seminar

Prof. Dr. Holger Giese, Christian Medeiros Adriano , Sona Ghahremani

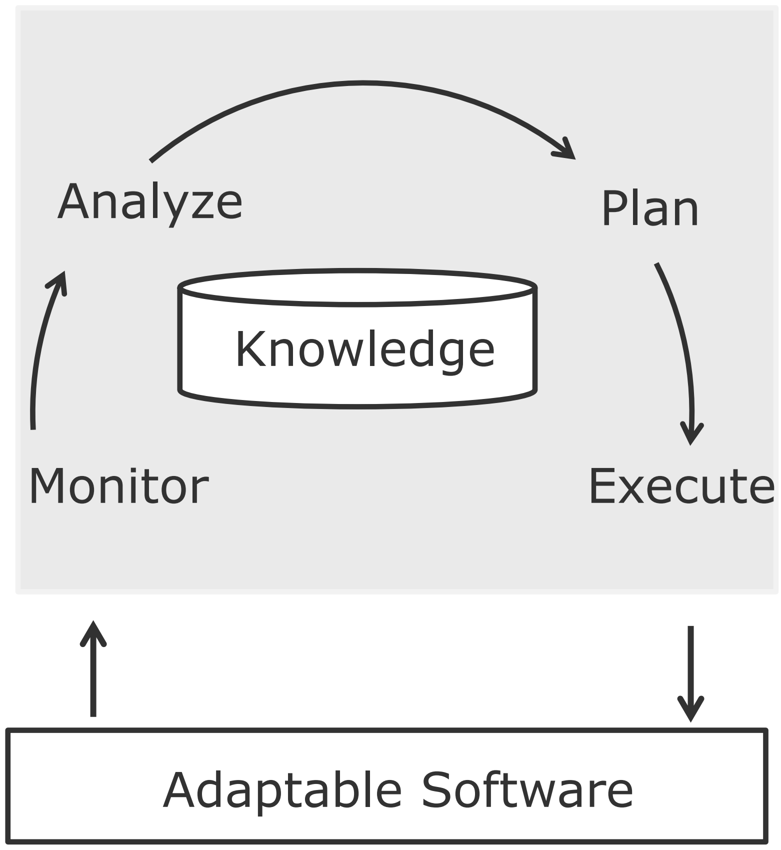
System Analysis & Modeling Group,
Hasso Plattner Institute for Digital Engineering
University of Potsdam, Germany

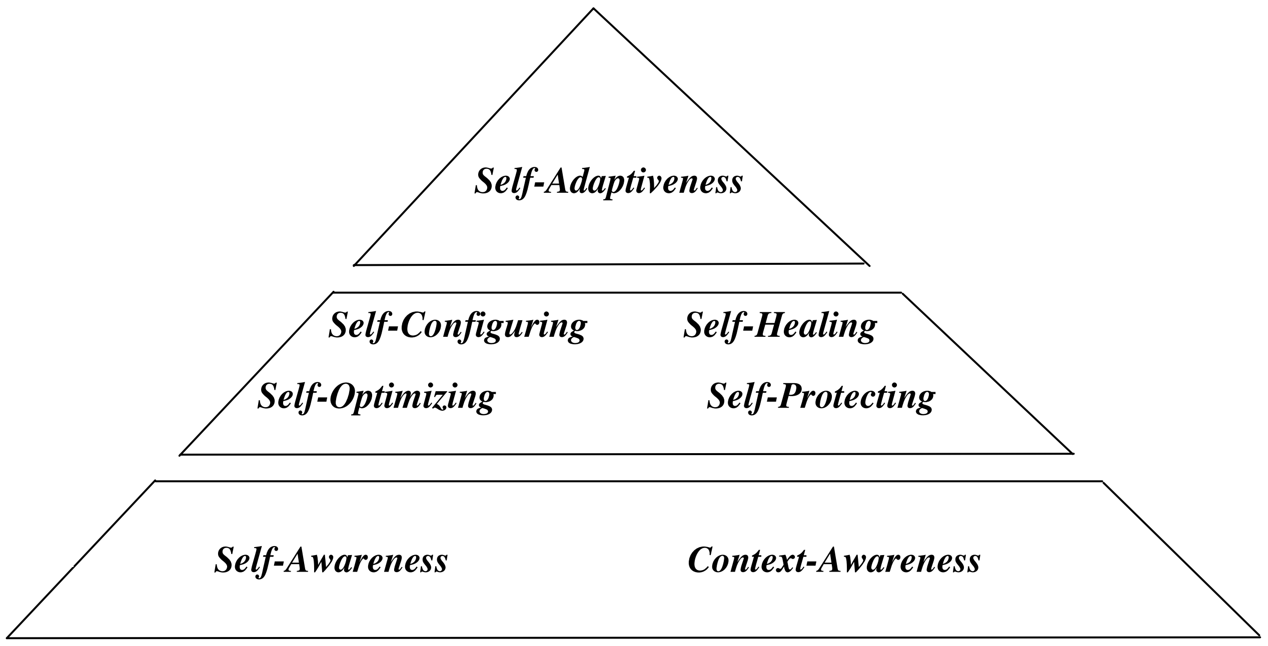
holger.giese@hpi.de
christian.adriano@hpi.de
sona.ghahremani@hpi.de

Sommersemester 2020

Hasso Plattner Institut
Digital Engineering • Universität Potsdam

Self-* Systems





Self-Healing and Self-Optimization Systems

Self-healing: is the capability of discovering, diagnosing, and reacting to disruptions at runtime.

Self-optimizing: is the capability of managing performance and resource allocation in order to satisfy the requirements.

Self-Healing Options

Any component can (1) crash, (2) throw exceptions, or (3) it can be destroyed (severe crash).

Possible countermeasures (possible actions or rules):

- light-weight redeployment of the component,
- restart the component,
- heavy-weight redeployment of the component, and
- replace a component with an alternative one, for instance, switch from internal authorization to an external authorization such as Facebook.

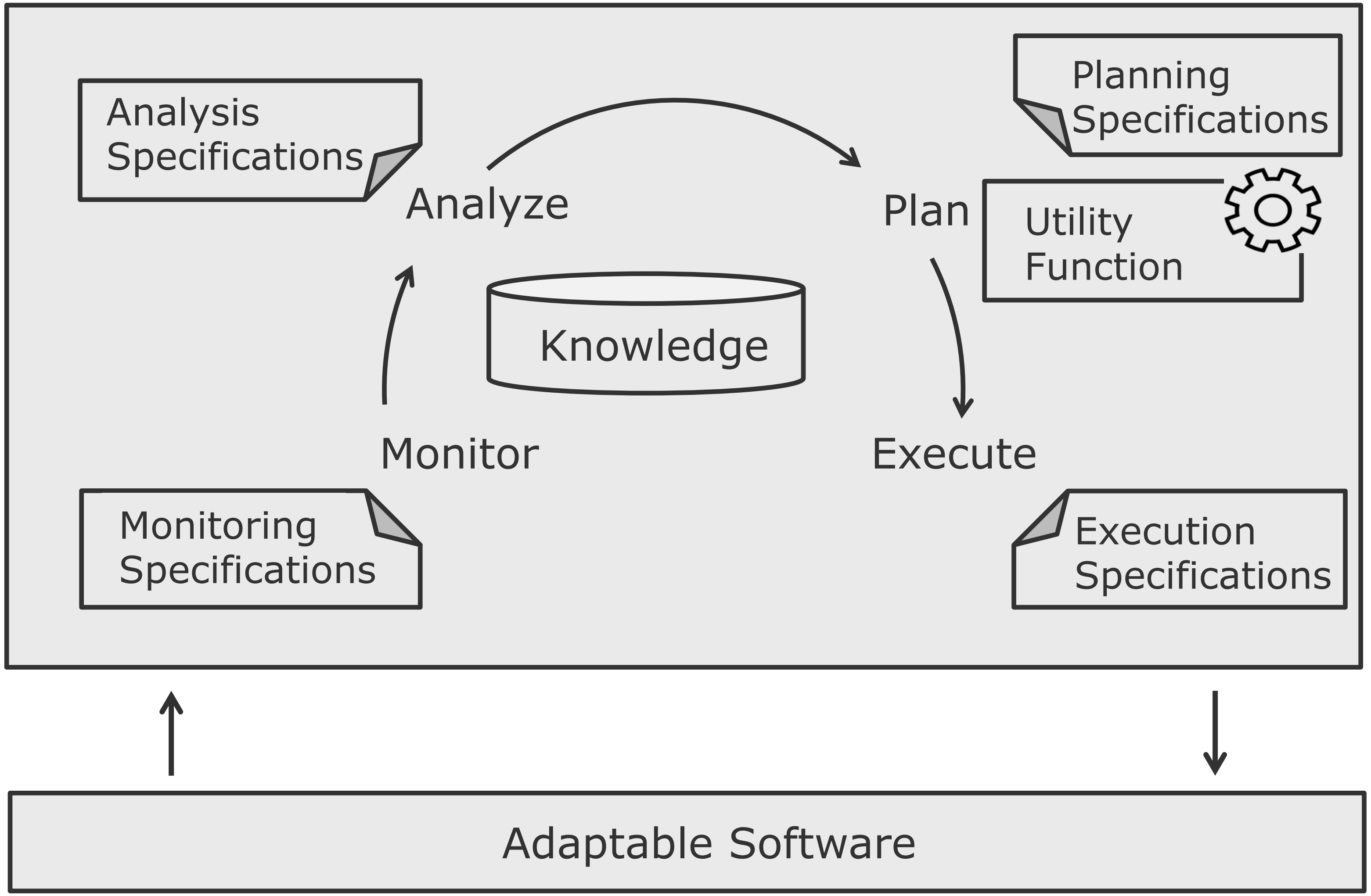
Self-Optimizing Options

The load of the system can vary and result in an increased or decreased demand for each service and its replicas.

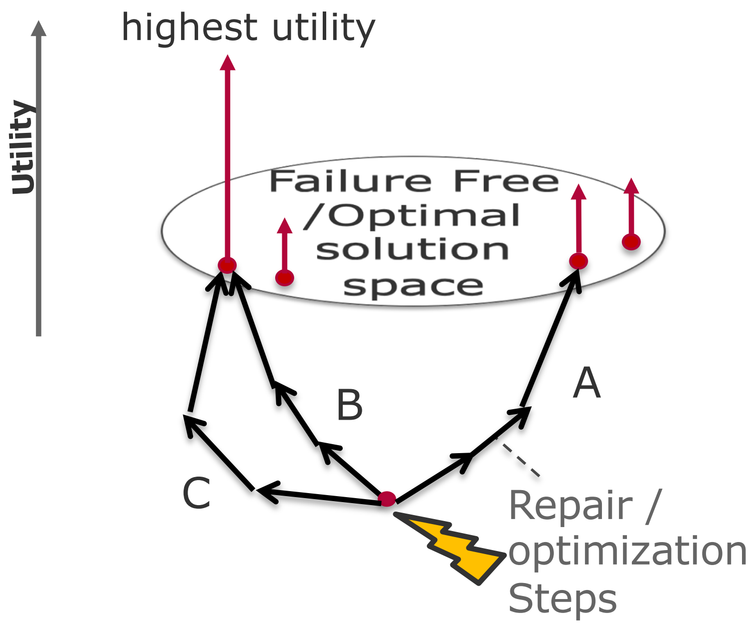
Possible countermeasures (possible actions or rules):

- Add replica to services in high demand
- Remove replica to reduce cost.

Self-* Systems : Specifications, Utility Functions



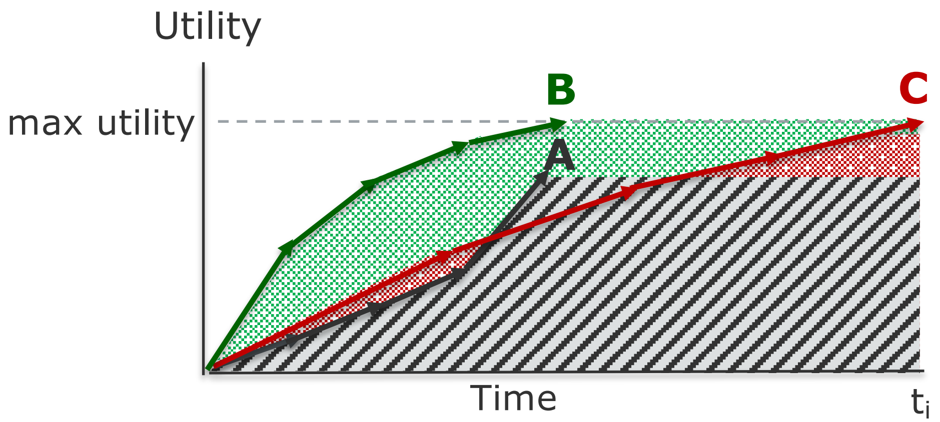
Adaptation Rules, Utility, and Reward



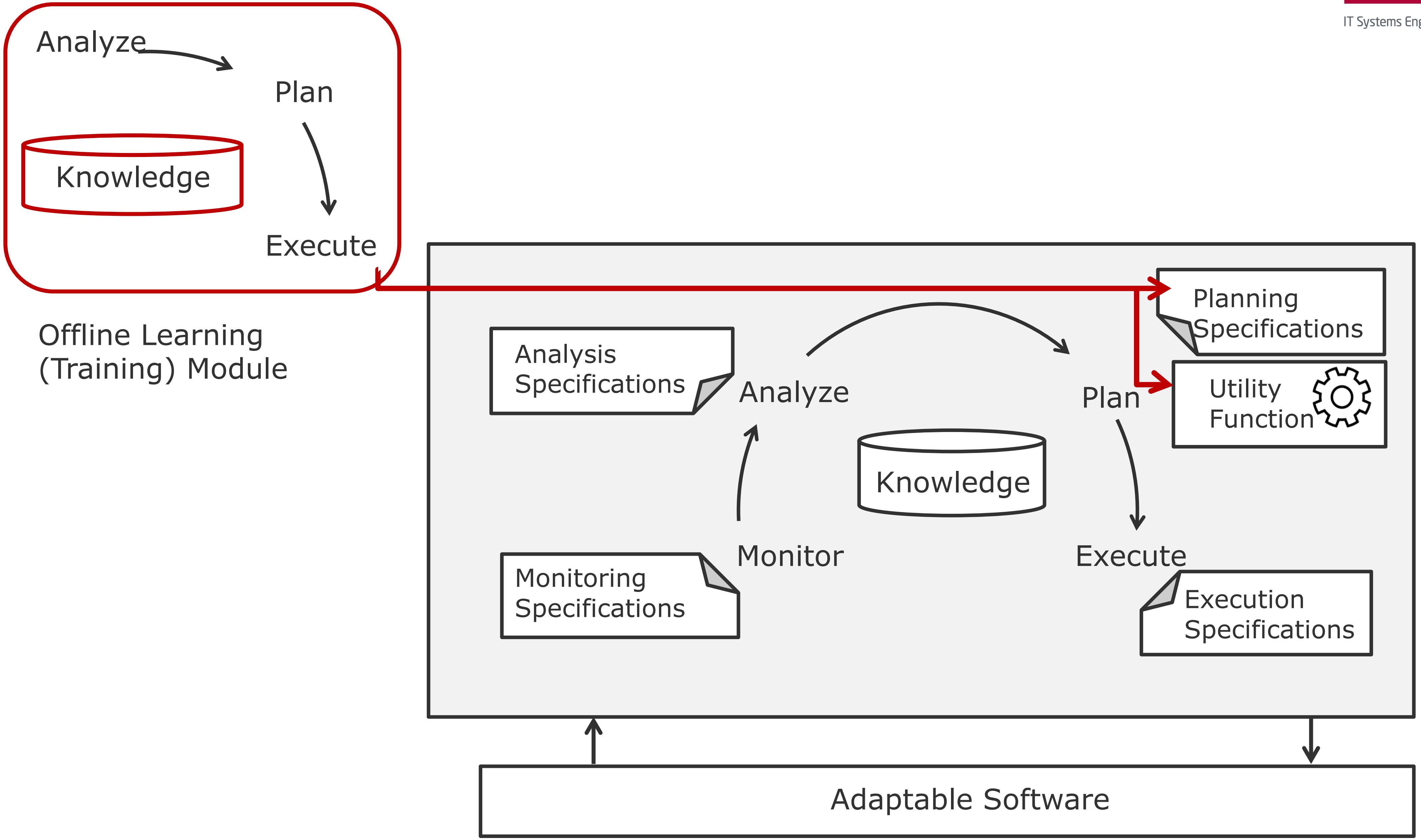
Sequence of repair/optimization steps -> **failure free / optimal solution space**

Final configuration with **highest utility**

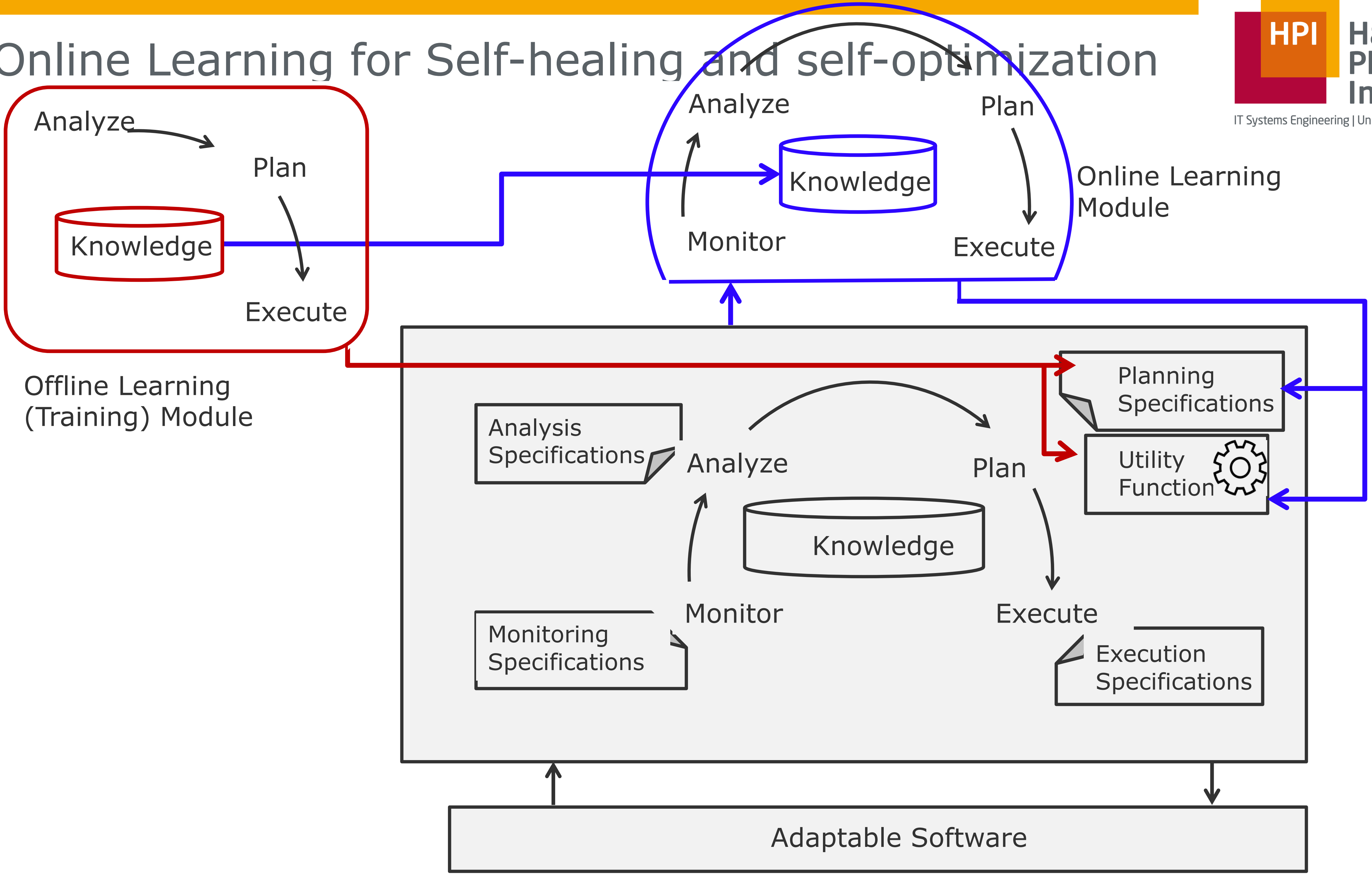
Optimal order of rules-> highest Reward



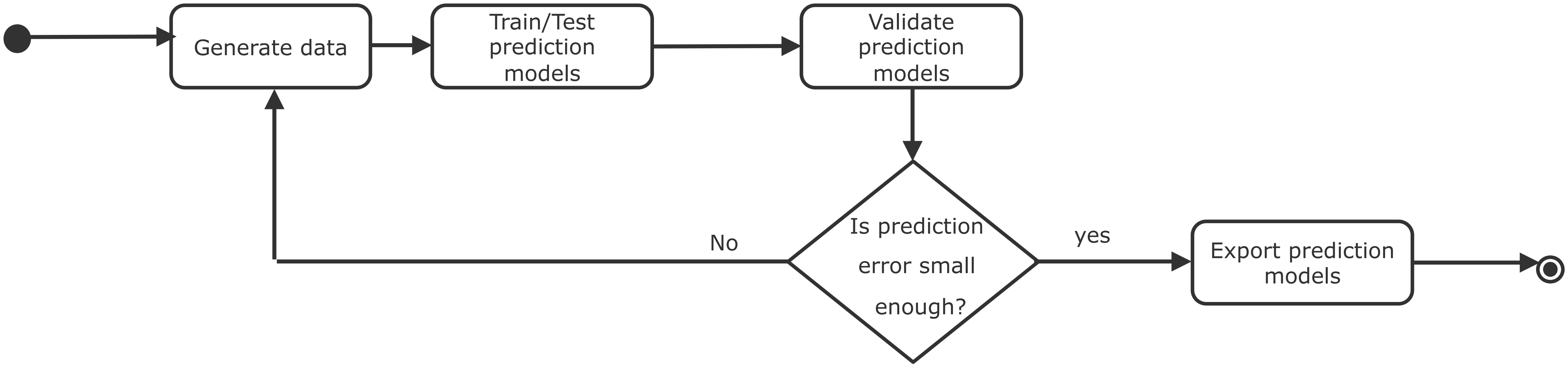
Offline Learning for Self-healing and self-optimization



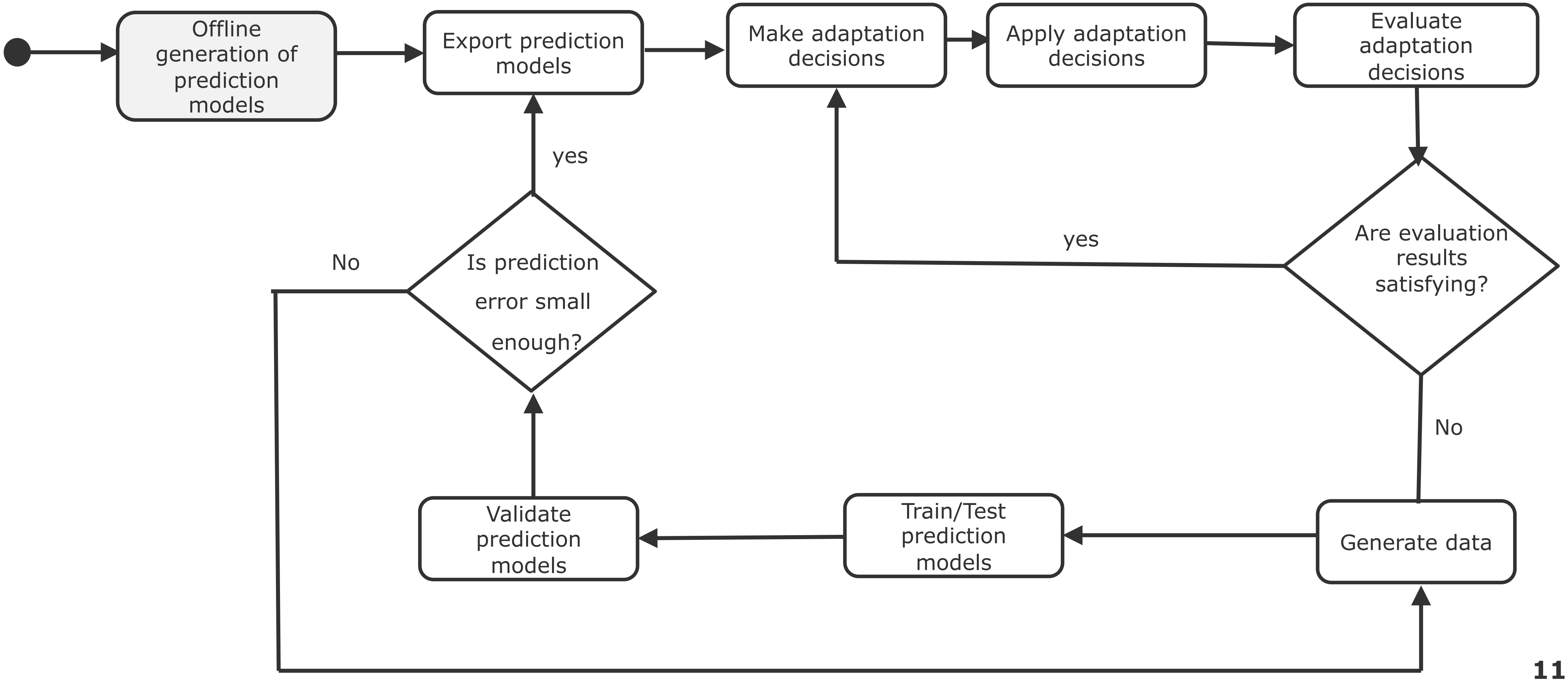
Online Learning for Self-healing and self-optimization



Offline Learning (detailed look)



Online Learning (detailed look)



Overview of Scenario and Models

Topics

Elements of Reinforcement Learning

- Exploration & Exploitation - Multi-armed bandits
- Markov Decision Process (MDP)
- Markov property, Bellman Equation, Optimality

Concepts and Theory and Assumptions

Model-Based methods

- Finite MDP
- Dynamic Programming (Policy and Value Iteration)

Model-Free methods

- Monte Carlo
- Temporal Difference (On-policy/Sarsa, Off-policy/Q-Learning)

Tabular Methods Finite/Small State Spaces

Approximate methods

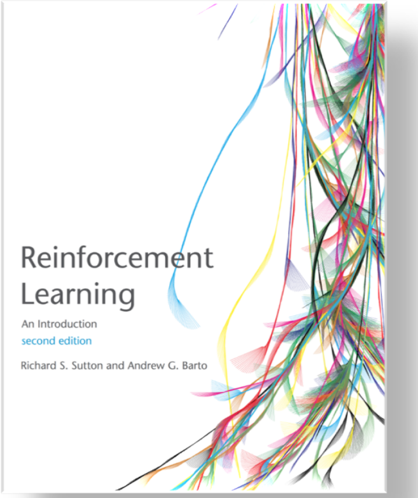
- Generalized solution (Incremental, Batch)
- Parameterized solution (Gradient Policy)

Function Approximation Infinite/Large State Spaces

Textbooks (free access)

Sutton, R. S., & Barto, A. G. (2015). ***Reinforcement learning: An introduction***. MIT press.

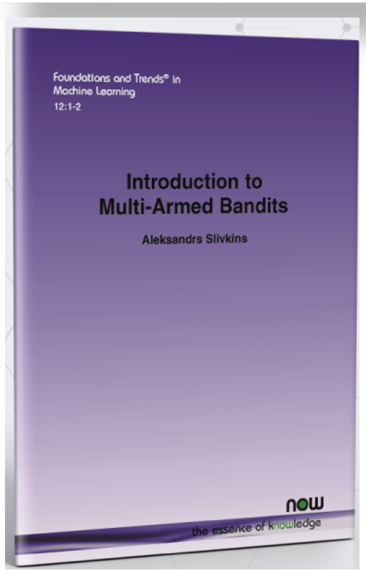
free access at the author's page: http://richsutton.com



Aleksandrs Slivkins (2019). ***Introduction to Multi-Armed Bandits***, Foundations and Trends® in Machine Learning: Vol. 12: No. 1-2, pp 1-286

free access by the authors'

https://arxiv.org/abs/1904.07272



Incremental implementations

Phase-1 Environment setup with Multi-armed Bandit

Phase-2 Model-Based implementation

Phase-3 Model-Free extension

Phase-4 Approximate model extension

"When solving a problem of interest, do not solve a more general problem as an intermediate step‟
Vladimir Vapnik [1]

General goal

For each phase, you will receive:

- a working algorithm that integrates with some simple environment
- a training version of the environment to train your algorithm
- a testing version of the environment to test your algorithm

Your goal will be to:

- Document the algorithm
- Make any needed extensions to integrate with new environment
- Run experiments and plot the results (error approximation by episodes)

Scenario

Event: One of more components failing

Adaptive response: Select the repair actions (rules) that more quickly restore the system

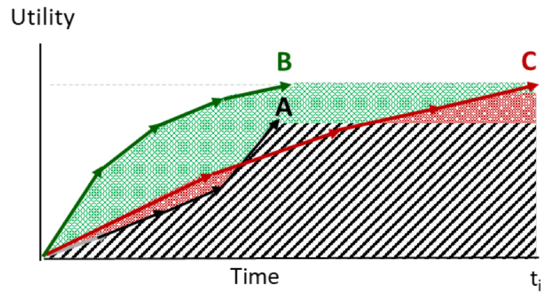
Challenges:

1. Multiple components might fail at the same time

2. For the same failure, there might be multiple actions that repair a component

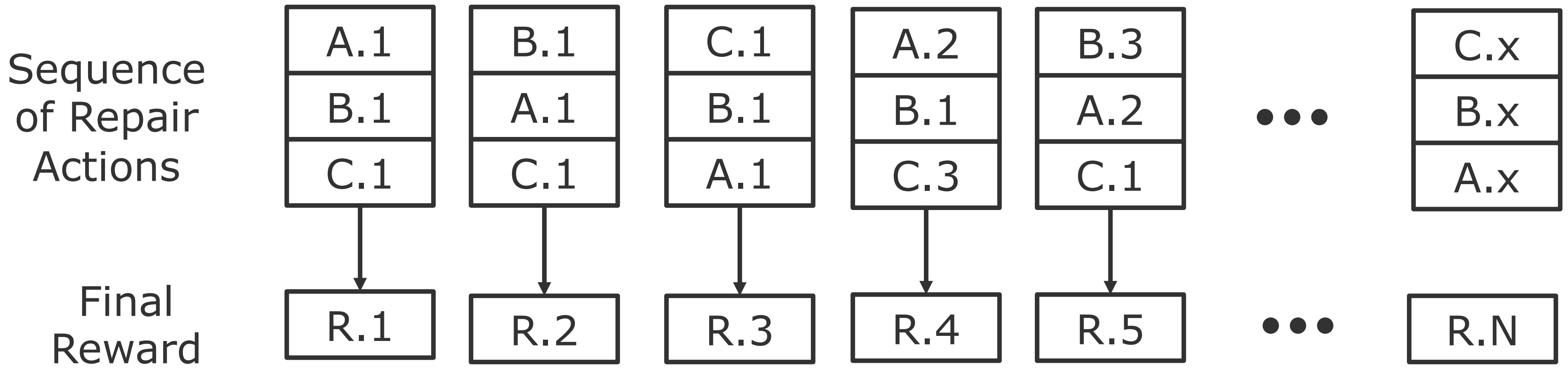
Goal:

Find the best sequence of repair actions

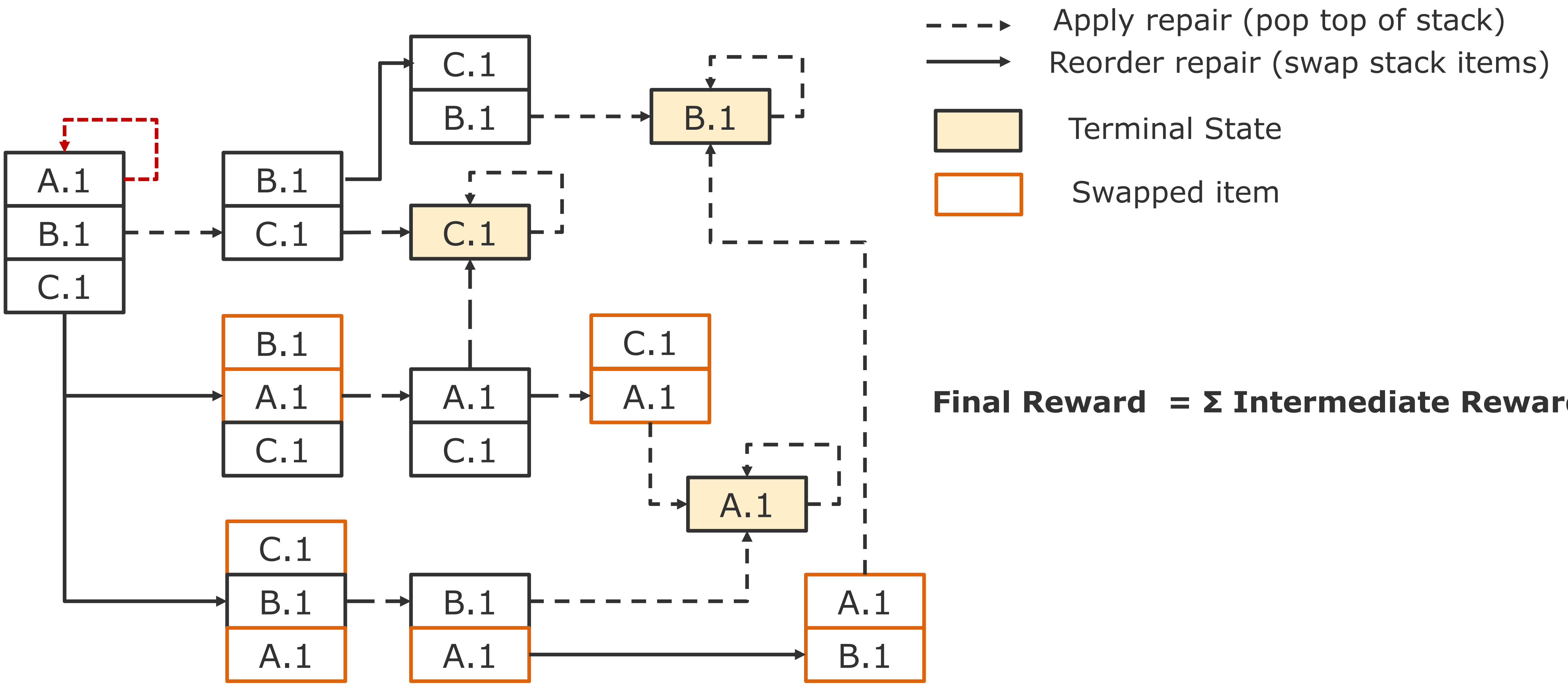


7

Multi-Armed Bandit Model



Markov Decision Process Model



Final Reward = Σ Intermediate Rewards

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

End