Towards flexible Runtime Monitoring Support for ROS-based Applications

Marco Stadler
Dept. of Business Informatics – Software Engineering
Johannes Kepler University
Linz, Austria

Michael Vierhauser
LIT Secure and Correct Systems Lab
Johannes Kepler University
Linz, Austria

Jane Cleland-Huang
Dept. of Computer Science and Engineering
University of Notre Dame
South Bend, USA

JOHANNES KEPLER UNIVERSITY LINZ

Runtime Monitoring

JOHANNES KEPLER UNIVERSITY LINZ

> "[…] monitors be installed to **gather and analyze** pertinent **information** about the **system's run-time** environment […] affect **adherence to requirements.**"

[Fickas and Feather 1996]

- **Often requires significant upfront investment**
- **Monitors need to co-evolve with the system**

Motivation - Runtime Monitoring for CPS

- Design and monitoring of CPS is recurring and crucial task in various domains

UAV search and rescue

CPPS

IoT

...

Motivation - Runtime Monitoring for CPS

- Close interaction between Humans and Hardware
- → requires runtime monitoring of the robot behavior
- → requires checking safety properties

The Problem I

Monitoring Solution A

Monitoring Solution B

Monitoring Solution C

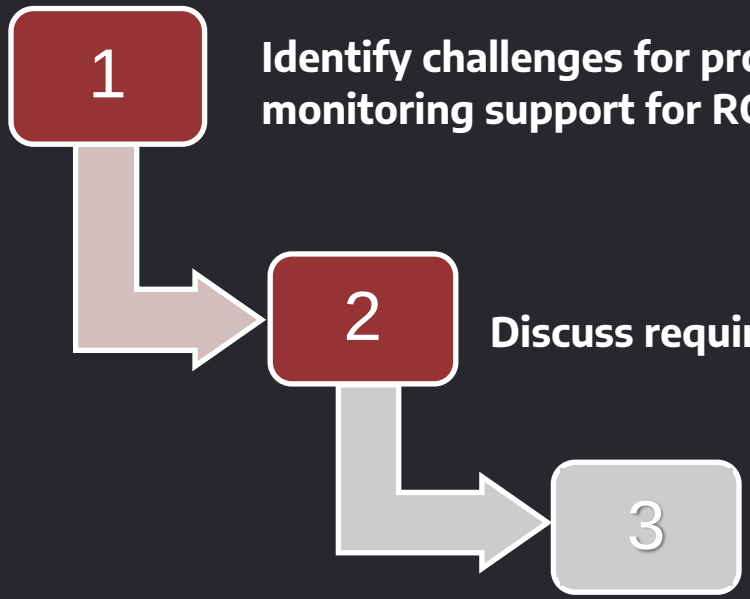
The Problem II

- **Reinventing the wheel for every new application**
  - Collection of data
  - Subsequent analysis
  - Checking functional behavior and (safety) constraints

Scope: **R**obot **O**perating **S**ystem

- Open-source software development platform for robotics applications and systems
- Many modern (industrial) robotic apps rely on ROS
- Steered by variety of industry partners including Amazon, Apex.AI, Bosch, Microsoft, Intel…

Approach



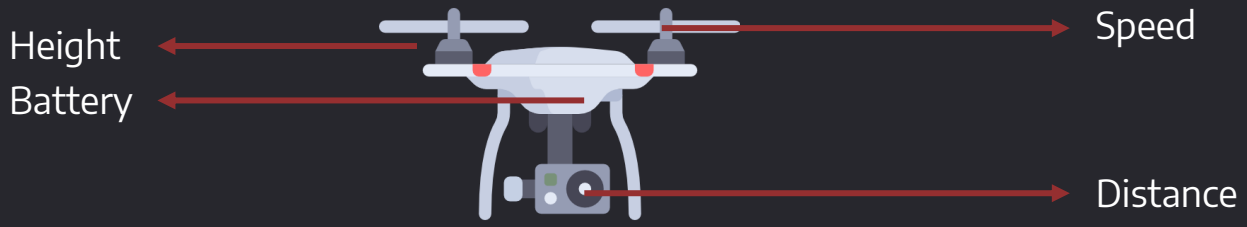
Identify challenges for providing effective monitoring support for ROS-based systems

Discuss requirements of such a monitoring framework

Implementation

Monitoring Challenges

1. Provision of initial system overview

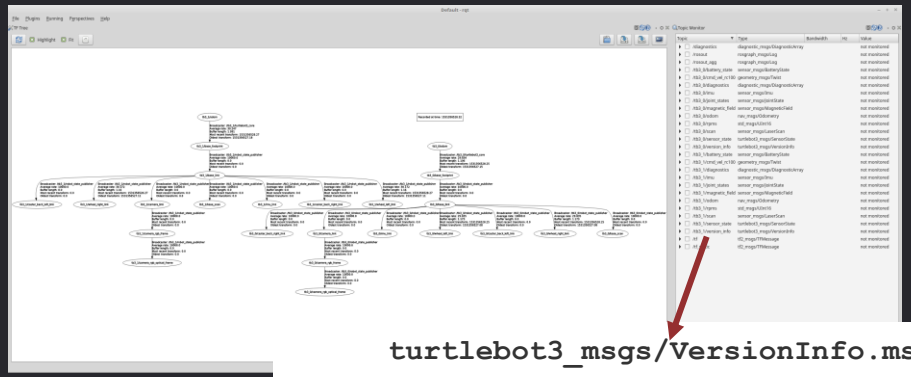


Monitoring Challenges

2. Diversity and individual monitoring needs

Monitoring Challenges

3. Only subset of properties is likely to be monitored continuously



turtlebot3_msgs/VersionInfo.msg

```
########################################
# Messages
########################################
string hardware   # .. : hardware version of Turtlebot3 (ex. 2021.05.23)
string firmware   # .. : firmware version of OpenCR
string software   # .. : software version of Turtlebot3 ROS packages
```

Monitoring Challenges

4. Adaptive monitoring

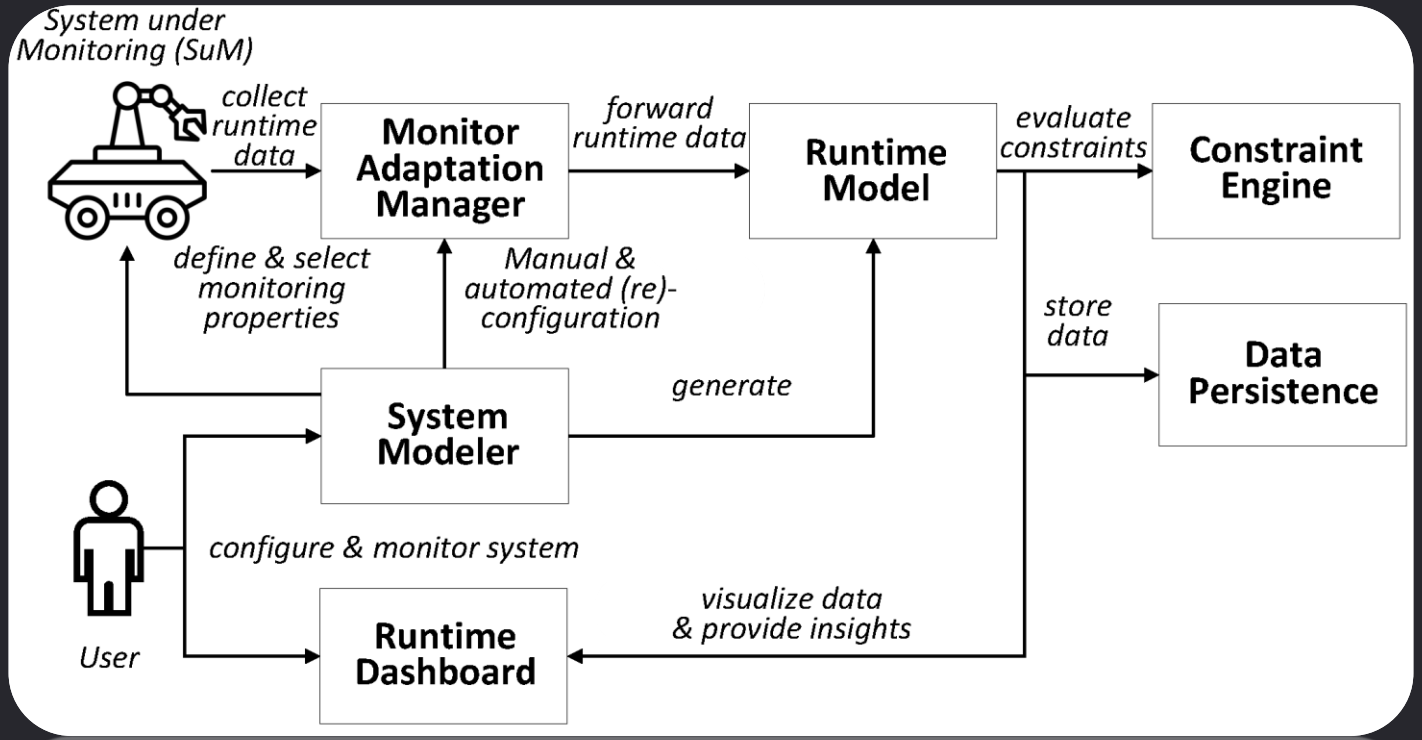
Monitoring Challenges

5. Constraint checks

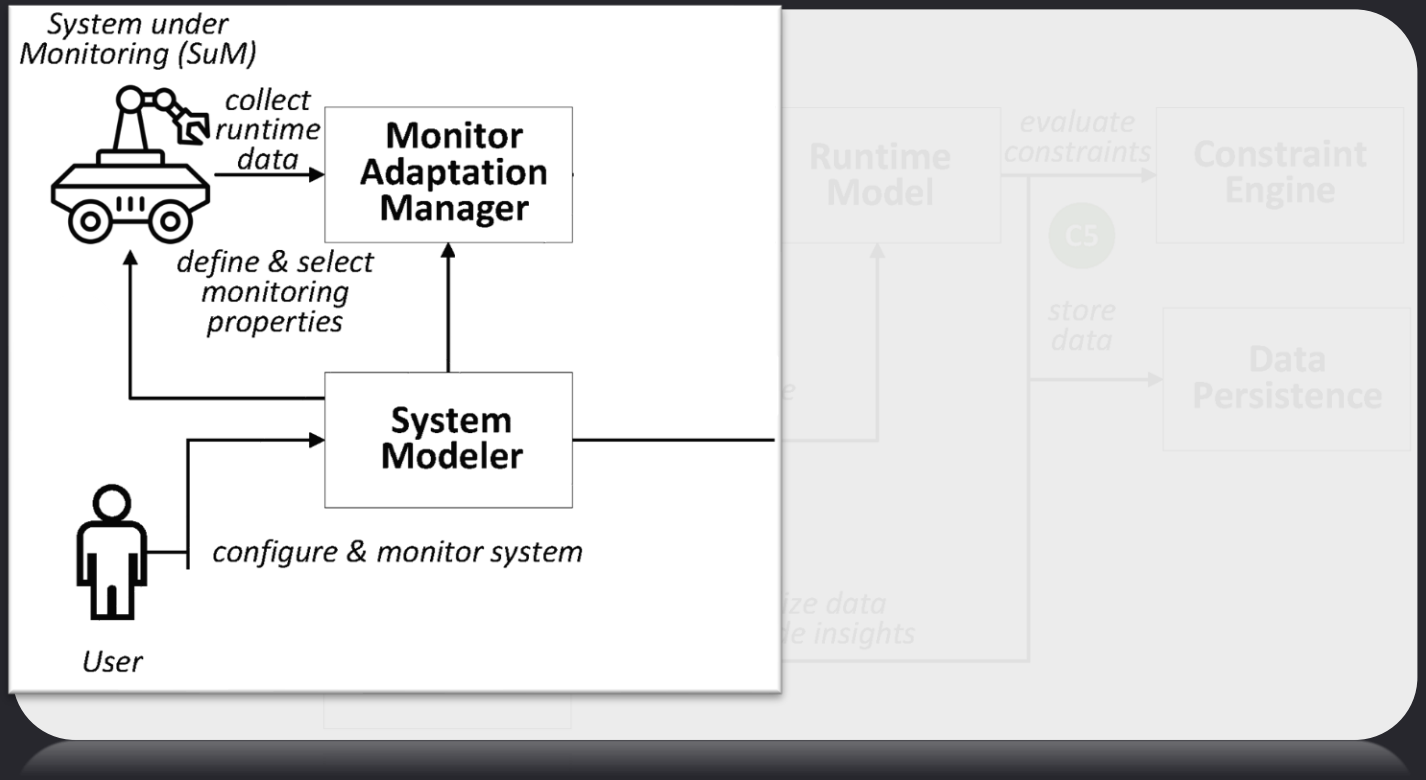
Monitoring Challenges

6. Runtime monitoring visualization

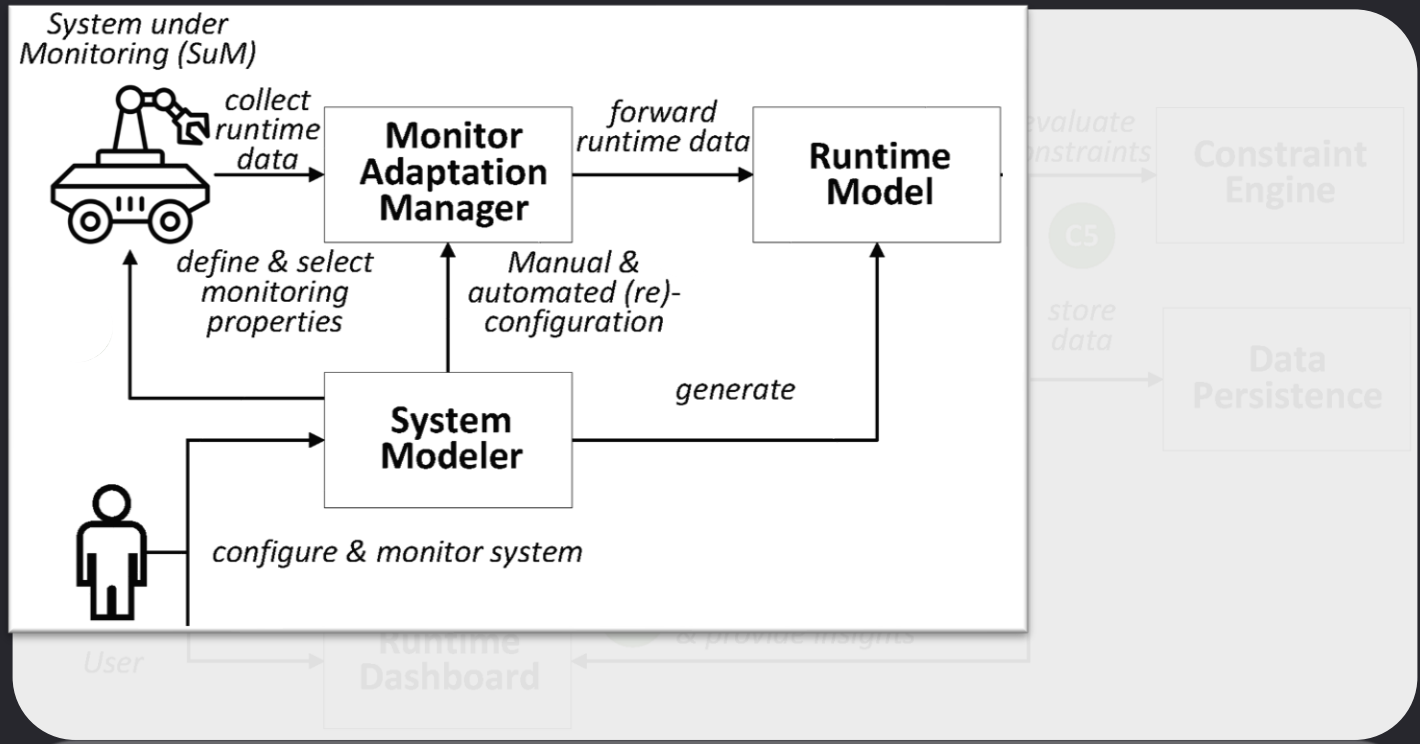
Approach: Architecture Overview



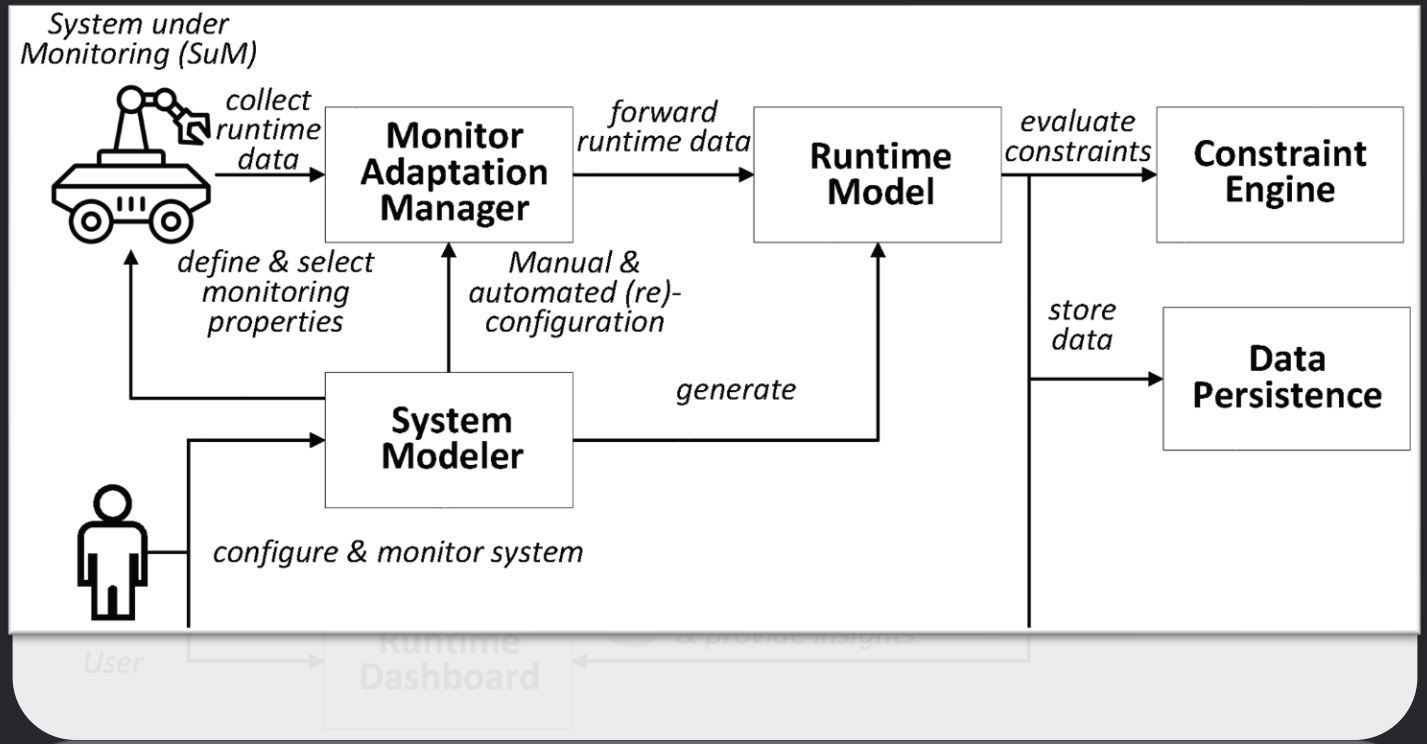
Approach: Architecture Overview



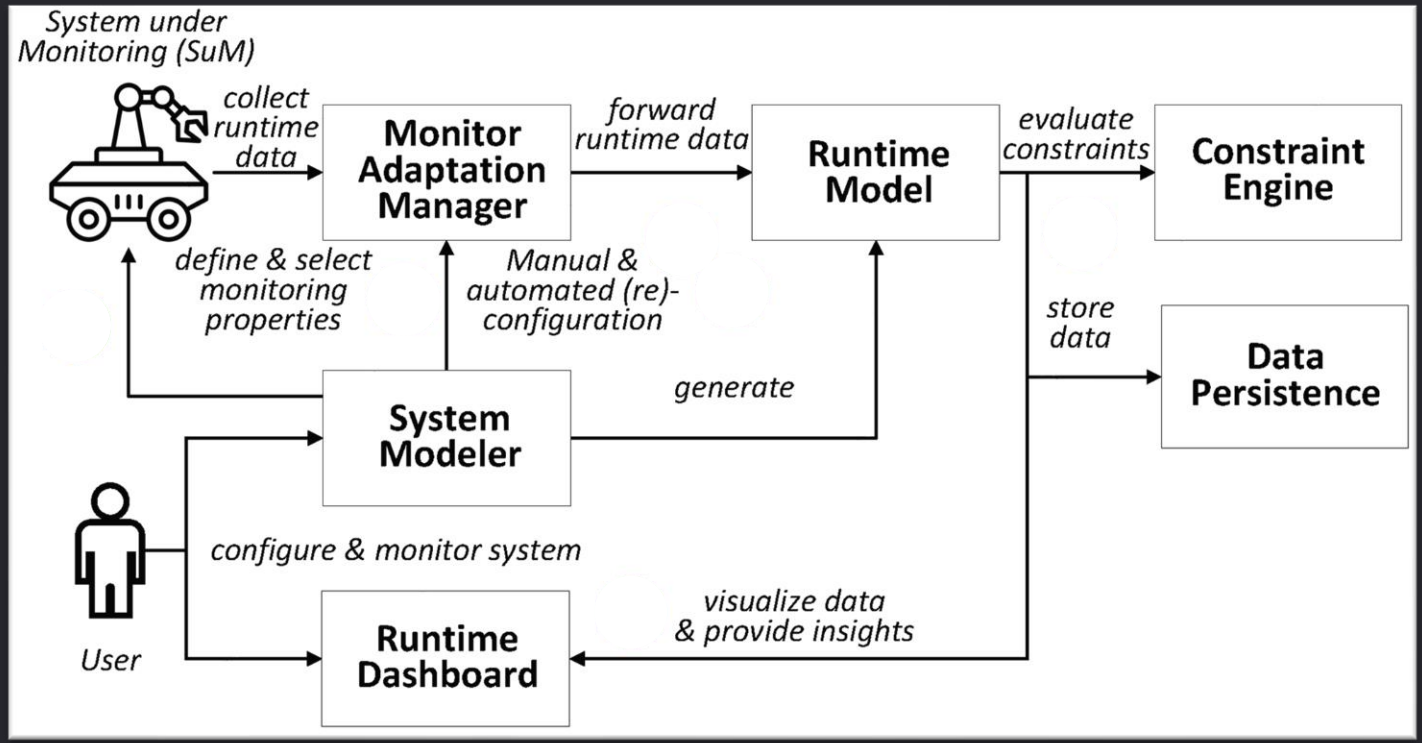
Approach: Architecture Overview



Approach: Architecture Overview



Approach: Architecture Overview



Conclusion

- **Design and deployment of a flexible and easily configurable monitoring framework for ROS-based systems**

- **First prototype**
  - Implemented Core-Features
  - Utilized Python ROS-bridge

Next Steps

- **Complete Implementation**
- **Evaluation with different ROS systems**
  - Drones
  - TurtleBots
- **Efficient Data Processing**

Marco Stadler
Dept. of Business Informatics – Software Engineering
Johannes Kepler University Linz
marco.stadler@jku.at