



The RoCS Framework to Support the Development of Autonomous Robots.

Leonardo Ramos*, Gabriel Divino, Breno B. N. de França, Leonardo Montecchi, Esther L. Colombini.

Abstract

With the expansion of autonomous robotics and the variety of applications found nowadays (e.g. medical, competition, military), the biggest hurdle on the development of mobile robots lies in endowing them with the capacity of interacting with the environment and making correct decisions so that their tasks can be executed successfully. Based on the wellknown IBM Autonomic Computing reference architecture (known as MAPE-K), this work defines a refined architecture following the Robotics perspective.

Key words:

Framework, Autonomous, Robotics.

Introduction

Typically operating in the real world it is expected that robots act through their perception, reasoning, planning, and decision-making process to accomplish their goals. However, as the complexity robotic systems grows and evolves, the need to organize and modularize software for their correct functioning also becomes more intricate and the available information becomes widely distributed on space and time, which drives the agent to search for relevant information.

Several architectures and frameworks that address these problems are available in the literature, but, despite all this, it is still an arduous and costly job to reuse software for robotic systems, either in a partial or complete form, when heterogeneous robot architectures and applications are considered.

The purpose of this work is to extend and adapt the Autonomic Computing Architecture defined by IBM [1] to comply with the current state of autonomous robotics, fostering reuse to support the development of autonomous robots through a development framework, called RoCS.

Results and Discussion

Based on the IBM's MAPE-K, we designed and implemented the RoCS Framework, a concrete framework to guide robot developers in structuring their code. The framework has been implemented in C++, which is one of the most popular languages in the robotics domain.

The core of the framework consists of one class for each of the five main blocks of the architecture: Monitor, Analyze, Plan, Execute, and Knowledge (Figure 1). The Monitor is responsible for the Observation step, Analyze for the Orientation, Plan for the Decision, and Execute performs the Action. Each of these blocks can be extended or rewritten so they can be tested individually, changing how the robot acts.

To evaluate the capability of the framework of supporting a hybrid approach, we defined a task where both deliberative and reactive routes can be activated. Hence, the robot task is to plan how to reach a specific position in the environment (origin) while avoiding collisions.

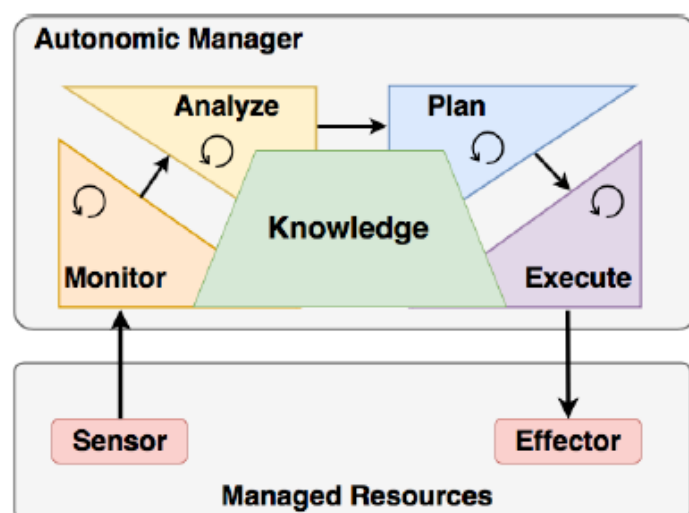


Image1. Autonomic Manager in the MAPE-K Architecture.

Conclusions

We presented an extension to the MAPE-K reference architecture towards the Robotics perspective, mainly for service robots applications with heterogeneous physical platforms. From this extension, we developed the RoCS framework to support the development of autonomous robots under a known architecture. We understand this approach can assist students or novices in robot development, and help experienced developers focus on their specific problems like machine learning algorithms for computer vision, sensor fusion techniques, and locomotion for robot models using particular physical devices.

Initially, the RoCS framework was evaluated in a simple but usual scenario for service robots as proof of concept. For future work, we envision the development of robots using the RoCS framework for competitions such as the RoboCup Humanoid Soccer Teen Size League and RoboCup Flying Robots Competition.

Acknowledgement

Authors thank to CNPq for the grant support to this research.

¹ Seyed Hossein Siadat, Minseok Song. 2005. An Architectural Blueprint for Autonomic Computing. Technical Report. IBM.