Introducing Q-Rock: Towards the Automated Self-Exploration and Qualification of Robot Behaviors

Thomas M. Roehr, Daniel Harnack, Oscar Lima, Hendrik Wöhrle
DFKI GmbH Robotics Innovation Center, Bremen, Germany
firstname.lastname@dfki.de

Frank Kirchner
DFKI GmbH Robotics Innovation Center, Bremen, Germany
University of Bremen, Germany
frank.kirchner@dfki.de

***Abstract*—The increasing complexity of robotic systems demands novel approaches for their design and special support for their development and programming. To tackle this issue, the project Q-ROCK presents a novel, life-cycle inspired development process for the automated self-exploration and qualification of robotic capabilities and behaviors. The development process in Q-ROCK consists of three interdependent steps: (1) the automated exploration of capabilities that given robotic hardware can perform, (2) the classification and semantic annotation of these capabilities to generate more complex behaviors, and (3) the mapping of existing application requirements to a robot system that is capable of fulfilling these requirements due to its set of capabilities and behaviors. Each of the steps builds upon the results of previous steps. Furthermore, Q-ROCK 's main intention lies in the introduction of a unified development cycle comprising the selection, development and improvement of robots, which can be used by robot experts and novice end-users alike. By traversing the development cycle multiple times, a comprehensive, continuously growing knowledge-base including but not limited to hardware and software components, their capabilities and inter-dependencies is evolved.**

***Index Terms*—automated robot design, behavior exploration, transfer learning, automated robot synthesis**

I. INTRODUCTION

Recent progress in the development of sensors, actuators and computing hardware enables the design of feature rich robotic systems. The robot Mantis [1], for instance, comprises 61 Degrees of Freedom (DoFs), and includes more that 400 sensors. At the same time, however, the amount of actuators and sensors presents a significant control, data processing, and finally a hardware and software integration challenge.

To facilitate the development of such systems in the future, an essential task is to effectively reduce the development complexity a robot designer has to handle. To achieve this, robot designers of different fields, e.g. mechanical and electrical engineers, software developers and machine learning experts, but also application-oriented end-users, need to be equipped with novel tools, for instance to select a robot for a specific task, to simplify the enhancement and improvement of existing robots, or to develop entirely new robots.

This research and development project is funded by the German Federal Ministry of Education and Research (FKZ 01IW18003)

There are already many tools available, for example [2] or robotic frameworks like ROS [3], Rock [4] that facilitate the general software integration for robotic systems, Computer Aided Design (CAD) software that facilitates the general mechanical design process, and machine learning libraries and frameworks like scikit-learn [5] or TensorFlow [6] that enable even inexperienced users to apply advanced artificial intelligence algorithms.

Although the existing set of tools is complementary, to our knowledge there is no unified and (semi-)automated approach permitting the co-development of hardware and software. Hence, the idea of Q-ROCK [7] is to apply artificial intelligence techniques to establish a development cycle for robotic systems that is driven by users' high level task specifications and requires minimal human intervention. To this end, the focus of Q-ROCK is the automated identification and design of robot *behaviors* by applying a holistic development process that integrates different perspectives.

In Q-ROCK we build upon the work of the predecessor Project D-Rock [8] that actually builds a hierarchical database of hardware components for robotic systems. The key feature of this hierarchy of D-Rock components is the fact that hardware and associated software are inherited upward through the hierarchy. D-Rock established a database of robotic components and created a workflow and supporting tools to model components, synthesize composite structures, and deploy corresponding hardware and software structure to control a target robot.

II. Q-ROCK DEVELOPMENT CYCLE

The development cycle is broken down into several steps and consider two main entry points into this cycle. The whole cycle including its entry points is illustrated in Figure 1. The first entry point (E1) permits the provisioning of a set

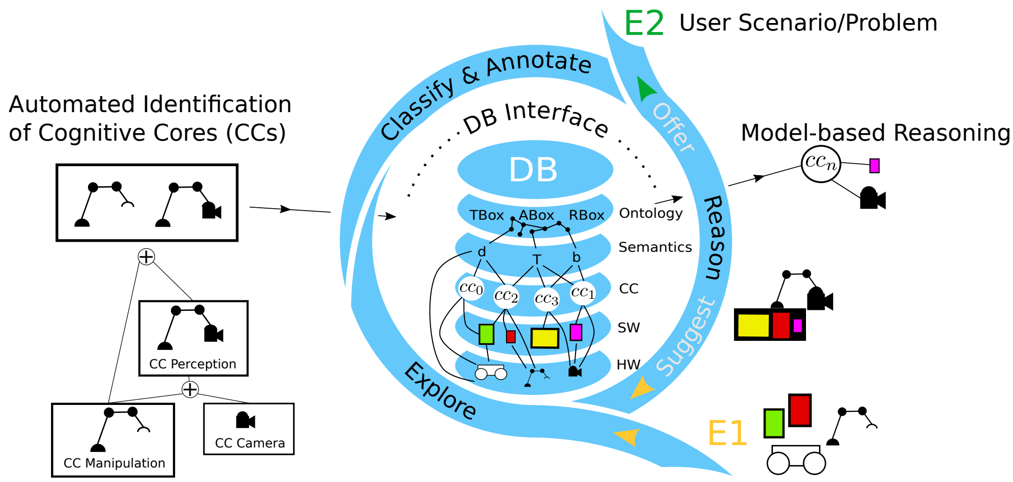

Figure 1: Q-Rock development cycle consisting of three interdependent stages and two entries points E1 and E2.

of available hardware and software components, for which capabilities are (at least partially) unknown and thus have to be explored.

In the *exploration phase*, a robotic system tries on its own different capabilities in a simulated environment, much like a newborn human or animal would in the real world. Here, the aim is to do this with as little goal direction as possible in order to generate a maximal variety of capabilities and to detect new, unanticipated capabilities. The results of this exploration – functions for actuating the robot's body and processing the sensory input – are robot experiences or rather *execution samples*. Execution samples are temporal sequences of robot states which are generated and stored in a database for later reference and analysis. These sequences represent the atomic capabilities of a robot. Based on these, composite and more complex behaviors can be generated at later stages.

Execution samples are forwarded to a *classification and semantic annotation phase* for analysis and characterization. This phase utilizes clustering approaches and embeds human support for the semantic annotation. During the clustering step, similar execution samples are grouped together and a centroid for each cluster is identified, so that it can serve as a representative for that cluster. In specific cases, these centroids are shown to human observers who can provide appropriate labels for annotation. These semantic labels are then attached to each cluster. The semantic annotation leads to so-called cognitive cores, which represent an essential building block in the Q-ROCK life-cycle, since they relate hardware and software components with appropriate control mechanisms and additionally provide a semantically meaningful and machine readable description of their behavior. Alternatively, semi-automated annotation can be built upon observations of human behaviour, e.g., when picking an object. Such observations can also be used as centroids for clustering.

A continuously growing database of cognitive cores forms the basis to support actual user queries, which represents the second entry point (E2) into the life-cycle. Given a user-specified problem description, the *structural reasoning phase* exploits these previously evolved cognitive cores to generate a suggestion of a robotic system. Note that the specific design of the system is unknown a-priori. For this, the problem is fist decomposed into a set of functional requirements. Based on existing semantic descriptions a set of applicable cognitive cores is queried from the database. The related hardware and software components can be identified and a structurally consistent system is assembled and suggested to the user as a potential solution to the specified problem.

REFERENCES

[1] S. Bartsch, M. Manz, and contributors, "Development and control of the multi-legged robot mantis," in *Proceedings of ISR 2016: 47st International Symposium on Robotics. (ISR-2016), June 21-22, München, Germany*. VDE VERLAG GmbH, 6 2016, pp. 379–386.

[2] G. S. Hornby, H. Lipson, and J. B. Pollack, "Generative representations for the automated design of modular physical robots," *IEEE transactions on Robotics and Automation*, vol. 19, no. 4, pp. 703–719, 2003.

[3] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," *ICRA Workshop on Open Source Software*, 2009. [Online]. Available: http://pub1.willowgarage.com/{~}konolige/cs225B/docs/quigley-icra2009-ros.pdf

[4] DFKI GmbH, "The Robot Construction Kit," 2011. [Online]. Available: http://www.rock-robotics.org

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] M. Abadi and Contributors, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[7] DFKI GmbH, "Q-Rock - the official web site," 2018. [Online]. Available: https://robotik.dfki-bremen.de/en/research/projects/q-rock.html

[8] ——, "D-Rock - the official web site," 2018. [Online]. Available: https://robotik.dfki-bremen.de/en/research/projects/d-rock.html