Chapter 11
Knowledge-Driven Autonomous Robotic Action Planning for Industry 4.0

Ajay Kattepur
Ericsson Research, India

ABSTRACT

Autonomous robots are being increasingly integrated into manufacturing, supply chain, and retail industries due to the twin advantages of improved throughput and adaptivity. In order to handle complex Industry 4.0 tasks, the autonomous robots require robust action plans that can self-adapt to runtime changes. A further requirement is efficient implementation of knowledge bases that may be queried during planning and execution. In this chapter, the authors propose RoboPlanner, a framework to generate action plans in autonomous robots. In RoboPlanner, they model the knowledge of world models, robotic capabilities, and task templates using knowledge property graphs and graph databases. Design time queries and robotic perception are used to enable intelligent action planning. At runtime, integrity constraints on world model observations are used to update knowledge bases. They demonstrate these solutions on autonomous picker robots deployed in Industry 4.0 warehouses.

INTRODUCTION

Advances in robotics, cyber-physical systems and industrial automation has come to the forefront with Industry 4.0 Lasi et al. (2014), with the following key requirements:

1. **Interoperability:** Machines, Internet of Things (IoT) Greengard (2015) enabled devices and humans connected and coordinating with each other.
2. **Information transparency:** Physical systems enhanced with sensor data to create added value information systems.
3. **Technical Assistance:** Use of intelligent devices to aid in informed decision making. Robotic automation may be identified to perform repetitive, unsafe or precise tasks.

DOI: 10.4018/978-1-7998-3375-8.ch011

4. **Decentralized Decisions:** The ability of such systems to make autonomous decisions; only critical cases will involve human intervention.

A fundamental characteristic required in Industry 4.0 deployments is the ability of autonomous robotic devices to self-configure in dynamic goal and deployment conditions. Autonomic computing Huebscher and McCann (2008) models have been proposed to create self-aware robotic systems that respond to both high level goals as well as external stimuli Faniyi et al. (2014). This has led to the development of *Cognitive Robotic Architectures* Levesque and Lakemeyer (2010)Beetz et al. (2010), that are at the intersection of robotics, IoT and Artificial Intelligence Russell and Norvig (2015).

Cognitive robots are able to intelligently execute tasks based on high level *goals*, dependent on *world model* knowledge and sensory *perceptions* to generate efficient *actions* Levesque and Lakemeyer (2010). In order to be deployed in dynamic Industry 4.0 environments, the robots must be autonomous and adaptive to runtime changes. Given a high level task such as "*pick ball from warehouse rack*", the autonomous robot must identify appropriate *action plans* to perform this task. As the robots are intended to be learning world models, *knowledge bases* are needed to populate information about the world, object, perception and action sequences needed. Any runtime anomalies are dealt with through further queries and eventual exception handling.

Distilling these high level requirements, an autonomous planning module for robots should include:

- *Knowledge Bases* that efficiently capture relationships between world models, objects, robot actions and tasks
- *Action Plans* that are efficiently decomposed from a high level goal task; this involves querying the knowledge base as well as triggering perceptions in case of knowledge mismatch
- Techniques to *Reconfigure* actions at runtime, when plans cannot be executed due to constraints
- Rules for consistent *Updates* to the world model, which allows multiple robots to coordinate or analyze exceptions during execution.

While individual modules may have been developed in the robotic and embedded software communities, integrating these features into a common framework for industrial deployments remains a challenge.

In this paper, we propose *RoboPlanner*, a structured technique to generate design time action plans for autonomous robots. In order to enable autonomy in deployments, we integrate *knowledge bases*, *design time action planning* and *runtime adaptation* modules. Knowledge representation and queries are enabled using efficient graph database technologies Angles and Gutierrez (2008). Design time action plans as provided using the formal concurrent programming knowledge Orc Kitchin et al. (2009), that allows structured composition of action plans. To take care of runtime adaptation, we provide general rules for triggering perception and exception handling. An integrity check is also provided to update the graph database with runtime knowledge. This framework is implemented over a realistic industrial use case involving autonomous picking robots employed in Industry 4.0 warehouses Wurman et al. (2008).

Principal contributions of this chapter:

1. *RoboPlanner* Knowledge Base module that formally models robotic world models, capabilities, object descriptions and task templates.
2. *RoboPlanner* Action Planner that uses design-time queries/updates to knowledge graph databases, including exception handling.

Related Content