

Eine Erweiterung für die konsistente Implementierung von Sensorik und Wahrnehmung im Plansys2 Framework

An Extension for Consistent Implementation of Sensors and Perception in the Plansys2 Framework

Simon Schläger*, M. Sc. schlaeger@igmr.rwth-aachen.de

Sören Walther**, M. Sc. s.walther@tu-braunschweig.de

Mathias Hüsing*, Prof. Dr.-Ing., Huesing@igmr.rwth-aachen.de

Burkhard Corves*, Prof. Dr.-Ing. Dr. h. c., Corves@igmr.rwth-aachen.de

*RWTH Aachen, Institut für Getriebetechnik, Maschinendynamik und Robotik, 52062 Aachen, Deutschland

**TU Braunschweig, Institut für mobile Maschinen und Nutzfahrzeuge, 38106 Braunschweig, Deutschland

Kurzfassung

Die steigende Komplexität moderner autonomer Robotersysteme erfordert eine in gleichen Maßen komplexere Systemarchitektur, die über traditionelle Automaten zur Aktionssteuerung hinausgehen und symbolische Planer zur Planung von Aufgaben miteinbeziehen [1]. Um die Komplexität und den Umfang der Systemarchitektur handhabbar und skalierbar zu halten, existieren Frameworks, die eine konsistente, wiederverwendbare und hinreichend abstrahierte Programmierung gewährleisten [1,7-8]. Dabei definieren diese Frameworks den Ansatz, nachdem die Systemarchitektur in einzelne Softwaresysteme unterteilt wird und auf welche Weise diese miteinander interagieren [2].

Bei der *Sense-Plan-Act* Architektur (SPA-Architektur) wird die Umgebung durch Sensoren wahrgenommen und durch Modellierung der Daten wird ein inneres Weltbild erstellt (*Sense*). Basierend darauf wird ein Plan zur Erreichung eines Ziels bestimmt (*Plan*), der dann umgesetzt wird (*Act*). Die SPA-Architektur zeichnet sich durch ihre deliberative Ausrichtung aus, was allerdings eine geringe Reaktivität bedeutet, da Veränderungen in der realen Welt keinen unmittelbaren Einfluss auf die Aufgabenausführung haben. Im Unterschied dazu werden bei verhaltensbasierten Ansätzen sensorische Inputs in Analogie zu menschlichen Reflexen direkt auf Aktionen abgebildet, was zu einer gesteigerten Reaktivität führt, jedoch auf Kosten der Fähigkeit deliberativ zu handeln. Eine mögliche Lösung besteht darin, hybride Architekturen zu verwenden, welche versuchen, die deliberative und reaktive Ausrichtung beider Ansätze zu vereinen, indem die Architektur aus einer Verhaltensebene und einer Planungsebene besteht. [2]

Mit *Plansys2* [1] existiert eine hybride Roboterarchitektur für die Aufgabenplanung und -ausführung, die das *Robot Operating Systems 2* (ROS2) [3] als Middleware verwendet. *Plansys2* verwendet auf der Planungsebene die *Planning Domain Definition Language 2.1* (PDDL) [4], um das interne Weltmodell zu erstellen und Planungsziele sowie planbare Aktionen zu definieren. Eine PDDL-Aktion modelliert eine reale, ausführbare Aktion unter Verwendung von Bedingungen, die erfüllt sein müssen, um die Aktion auszuführen. Zusätzlich dazu beschreibt sie Effekte, die durch die Ausführung der Aktion hervorgerufen werden und zu Veränderungen im inneren Weltbild führen. Eine Abfolge von planbaren Aktionen bildet dann den Plan, welcher in der Verhaltensebene mithilfe von *Behaviour Trees* (BTs) [5] verhaltensbasiert ausgeführt wird. BTs sind gewurzelte Bäume, bei denen die äußeren Knoten Aktions- und Bedingungsknoten darstellen. Ein Tick-Signal wird über die inneren Knoten durch den Baum geleitet und steuert so den Ablauf in Abhängigkeit von den Rückgabewerten der äußeren Knoten (*Running*, *Success* oder *Failure*). Als Bindeglied zwischen der Planungs- und Verhaltensebene nutzt *Plansys2* das eigens entwickelte *Action Auction Protocol*, welches für jede geplante PDDL-Aktion, die dazugehörige implementierte ausführbare Aktion aufruft.

Die Besonderheit von *Plansys2* liegt in der konsequenten Verwendung von BTs. BTs werden sowohl auf hohem Abstraktionsniveau eingesetzt, um die Aufgabenausführung hinsichtlich Parallelitäten zu optimieren, als auch auf niedrigem Abstraktionsniveau, um einzelne ausführbare Aktionen zu implementieren. Neben BTs bietet *Plansys2* noch die Möglichkeit, ausführbare Aktionen mithilfe von ROS2-*Actions* zu implementieren. Die Einbindung von Sensoren wird in *Plansys2* kaum unterstützt. Aktuell ist keine Lösung bekannt, die eine Integration von Sensoren in die Planungs- oder Verhaltensebene von *Plansys2* mittels eines Frameworks vereinheitlicht. In Zukunft sollen die Schließung dieser Lücke angestrebt und eine konsistente Einbindung von Sensoren in *Plansys2* erreicht werden.

Zunächst soll sichergestellt werden, dass Sensoren im *Plansys2*-Framework einheitlich auf verschiedenen Abstraktionsebenen programmiert werden können, um sowohl Sensoren in BT-Bedingungsknoten als auch Sensoren zur Erstellung des inneren Weltmodells zu unterstützen. Zu überprüfen dabei ist, ob Stenzels [6] Betrachtungsweise von abstrakten Sensoren in diesem Kontext anwendbar ist. Darauf aufbauend wird eine Methode entwickelt, um die Bedingungen einer PDDL-Aktion zur Laufzeit konkret zu überprüfen, indem BT-Bedingungsknoten verwendet werden. Das Ziel dieser

Implementierung ist eine für den Anwender nachvollziehbare Ausführung eines Plans, um im Fall eines gescheiterten Planes Gründe hierfür zu identifizieren. Zusätzlich wird angestrebt, dass durch diese engere Verknüpfung von Sensoren zwischen der Planungs- und Verhaltensebene das Problem inkonsistenter Zustände bei einem Abbruch der Aufgabenausführung beherrschbarer wird. Inkonsistente Zustände treten auf, wenn das interne Weltmodell aufgrund der Anfangseffekte einer PDDL-Aktion aktualisiert wird, aber seine Aktualisierung aufgrund der Schlusseffekte einer PDDL-Aktion wegen des Abbruchs nicht stattfindet [1].

Literatur

- [1] F. Martín, J.G. Clavero, V. Matellán, F.J. Rodríguez, *PlanSys2: A Planning System Framework for ROS2*, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.
- [2] B. Siciliano, O. Khatib, *Springer Handbook of Robotics*, 2nd edition, Springer, Heidelberg, 2016.
- [3] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, *Robot Operating System 2: Design, architecture, and uses in the wild*, Science Robotics vol. 7, Mai 2022.
- [4] M. Fox, D. Long, *PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains*, Journal of Artificial Intelligence Research, Dezember 2003.
- [5] M. Colledanchise, P. Ögren, *Behavior Trees in Robotics and AI*, CRC Press, 2018.
- [6] R. Stenzel, *Steuerungsarchitekturen für autonome mobile Roboter*. Aachen, Techn. Hochsch., Diss., 2002.
- [7] M. Mayr, F. Rovida, V. Krueger, *SkiROS2: A skill-based Robot Control Platform for ROS*, 2023.
- [8] M.Á. González-Santamarta, F.J. Rodríguez-Lera, C. Fernández-Llamas, V. Matellán-Olivera, *MERLIN2: MachinEd Ros 2 pLanINg*, Software Impacts. 15 (2023).

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub | universitäts
bibliothek

In: Zehnte IFToMM D-A-CH Konferenz 2024

Dieser Text wird via DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt. Die hier veröffentlichte Version der E-Publikation kann von einer eventuell ebenfalls veröffentlichten Verlagsversion abweichen.

DOI: 10.17185/duepublico/81692

URN: urn:nbn:de:hbz:465-20240304-105223-2



Dieses Werk kann unter einer Creative Commons Namensnennung 4.0 Lizenz (CC BY 4.0) genutzt werden.