
使用面向对象编程实现通用机器人技能

A PREPRINT

Abdullah Farrukh

Innovative Fabric Systems

German Research Institute for Artificial Intelligence

Trippstadterstrasse 122, Kaiserslautern 67663

abdullah.farrukh@dfki.de

Achim Wagner

Innovative Fabric Systems

German Research Institute for Artificial Intelligence

Trippstadterstrasse 122, Kaiserslautern 67663

achim.wagner@dfki.de

Martin Ruskowski

Innovative Fabric Systems

German Research Institute for Artificial Intelligence

Trippstadterstrasse 122, Kaiserslautern 67663

martin.ruskowski@dfki.de

August 15, 2025

ABSTRACT

使用面向对象编程实现通用机器人技能

Keywords object-oriented programming · robot skills · interoperability

1 引言

研发机器人算法并将机器人子系统集成到更大型系统中可能是一项困难的任務。特别是在机器人技术知识匮乏的小型和中型企业 (SMEs) 中, 实施、维护和开发机器人系统是一个挑战。因此, 许多公司依赖系统集成商的外部专业知识, 而这在某些情况下可能导致供应商锁定和对外部的依赖。在智能制造系统的学术研究中, 机器人在设计稳健的自主系统中起着关键作用。研究人员在想要将机器人系统作为更大智能系统的一部分使用时, 也面临着类似的挑战, 而不想处理机器人接口的复杂性和广泛性。本文提出了一种软件框架, 可以减少部署一个可用机器人系统所需的工作量。重点是提供一个简化现代机器人系统不同接口的概念, 并使用一个针对不同厂商和型号的抽象层。使用 Python 编程语言实现了这一概念的原型。目标系统是一个包含 Yaskawa Motoman GP4 的分拣单元。

2 相关工作

类似的方法在过去已经被追求、实施并成功采用。机器人操作系统 (ROS) [1] 及其实时继任者 ROS2 [2] 在机器人社区中广为人知, 是开发机器人应用程序的首选框架, 特别是在研究领域。尽管 ROS 的使用提高了机器人软件的模块化和可用性, 但它仍然需要特定框架的知识才能让用户开发自定义应用程序。机器人库是一个 C++ 框架, 使用面向对象的软件架构来实现机器人应用的开发 [3]。作者在一个与平台无关的库中提供了一整套工具, 从数值方法到路径规划。作者提供了一种针对机器人以及传感器、夹持器等设备的硬件抽象层的自定义实现。开放机器人和动画虚拟环境 (OpenRAVE) 是另一个用于开发机器人应用的强大框架 [4]。与 ROS 类似, OpenRAVE 具有模块化结构并且可扩展, 可以与 ROS 和 Player [5] (另一个机器人软件框架) 一起使用。在撰写本文时, OpenRAVE 提供了 C++ 和 Python API。在工业中, 标准机器人命令接口 (SRCI) 是一个新兴的框架, 用于通过可编程逻辑控制器 (PLC) 控制工业机器人 [6]。基于服务器-客户端架构, 目的是实现硬件无关的应用设计, 类似于前面提到的方法。PLCOpen 开放标准是另一种在工业中创建可重用应用代码的方法 [7]。我们的方法专注于非领域用户, 并使用预先存在的硬件抽象在其上构建应用层。此外, 上述方法没有完全考虑到当前和未来工业 4.0 领域的要求, 我们旨在解决这些要求, 以实现一个完全自主的机器人系统。

在这项工作中，我们旨在创建一个框架以减少研究中部署机器人应用的时间。为此，我们首先收集研究部门中使用的各种机器人系统的需求。这包括关于机器人制造商和型号、所需的控制接口和中间件，以及应用类型的信息，例如拾取和放置任务。为了符合工业 4.0 概念，附加的需求包括使用资产管理外壳 (AAS) 和基于开放平台通信统一架构 (OPC UA) 的标准化控制接口。AAS 技术用于以标准化和互操作的方式描述机器人系统的配置。该配置被用作自动化设置机器人系统的输入。在收集需求后，使用类图对软件架构进行建模，并应用面向对象编程的基本规则和概念。然后，使用 Python 编程语言实现所建模的软件架构，并在一个料箱拣选单元上进行测试。由于机器人系统可以采用不同形式，本文仅关注具有 6 个自由度的工业机器人手臂。

3 需求

通过分析代码库中现有软件和采访过去曾与工业机器人合作、目前正在开发机器人应用或将在即将到来的项目中使用机器人的用户，收集了需求。向工业 4.0 领域的领域专家询问了对标准化控制接口的需求。总之，收集的需求可以分为三个主要类别：可用性、兼容性和互操作性：

i) 可用性

可用性要求解决了用户在机器人领域知识上的差异。内部研究表明，大多数机器人用户并没有机器人系统的深入知识，只是将其作为子系统或组件来实现更高级别的自动化。因此，该框架需要具有一个用户友好的 API 来设计用户应用程序。假设该框架的所有用户都已经获得了使用机器人系统的一般安全说明。

ii) 兼容性

确保与其他软件模块的兼容性是软件框架的关键要求。此功能允许机器人系统集成到一个更大、更复杂的自动化系统中，该系统通过多智能体系统 (MAS) 对硬件模块进行自主控制。同时，对控制 API 的高可靠性和可用性有要求。为了更好地应对不同制造商和型号接口及中间件的高度差异性，应该以模块化的方式重用现有的、制造商提供的接口。软件库中的现有应用程序主要使用机器人操作系统 (ROS) 框架。其他应用程序使用专有接口，例如 Universal Robots 的实时数据交换 (RTDE)。当前和未来的项目已经使用或计划使用需要集成到接口层的 OPC UA 接口。此外，作为软件架构的基本要求，每个制造商和他所提供的机器人的接口必须以模块化和可扩展的方式进行集成。由于仿真在机器人应用程序开发中起着关键作用，必须在框架设计中考虑对仿真框架的支持。

iii) 互操作性

为了使非专业人士能够设计机器人应用程序或将机器人系统作为子系统进行集成，互操作性被提取为关键需求。为实现这一目标，软件框架需要一个硬件，并且如果硬件或软件限制允许的话，要求实现一个对接口无关的基本机器人技能集，并提供添加额外、特定应用技能的方法。如果机器人应用程序要用作系统的独立组件，则外部控制 API 必须确保一个广泛采用的标准。在作为应用程序组件集成的情况下，例如 OPC UA 服务器，适用“良好”软件工程的要求。此外，必须使用最先进的语义技术 AAS 来描述机器人系统的配置，以确保对系统的共同理解。这包括有关机器人的基本信息 (制造商、型号等)、其接口 (例如 ROS、OPC UA 等) 以及接口特定的信息，例如主题名、可用服务等，这些信息需既可机器读取也可人工读取。

4 框架概念的提出

考虑到上述要求和现有方法，多层架构似乎是一种最有前途的方法。通过将硬件及其抽象与应用层和控制层分离，可以保证所设计算法的高度可重用性。此外，还实现了与硬件无关的机器人功能的实施，这进一步提高了机器人软件的可重用性并减少了部署工作量。我们方法中的硬件和接口抽象层使用预先存在的软件来控制机器人系统并访问其参数。这些实施由供应商维护，确保兼容性和长期支持。在我们研究部门的机器人系统中，ROS 支持是可用的。此外，大多数工业机器人都提供原生接口和 OPC UA 接口。挑战在于适应 Python 库的控制 API，以适应不同制造商和可用抽象，这通过使用 Python 的可扩展模块设计来实现，以确保应用层的应用程序能够独立于硬件运行。

4.1 软件架构 & 实现

为了使用面向对象编程实现该概念，设计了一个用于 Python 库初始原型的类图，如 2 所示。抽象类用于为每个类提供设计规范，通过继承使具体实现既一致又灵活。RobotControl 类是主要组件，它提供了与硬件无关的控制 API 结构。Manufacturer 类用于映射特定硬件和接口的过程和功能。组合允许在 Manufacturer 类中添加多个接口，并且在运行时只有一个接口处于活动状态。每个制造商都可以提供多个接口来控制机器人系统。抽象类 RobotInterface 用于实现不同接口，并将目标机器人系统的控制 API 映射到制造商类。由于技能可以具有不同的范围并包含不同的功能，抽象 Skill 类为 RobotSkill 类提供基本结构。每个定义的机器人技能通过使用 RobotControl 类的方法来实现 RobotSkill 类，从而允许硬件无关的算法开发。在我们的工作中，我

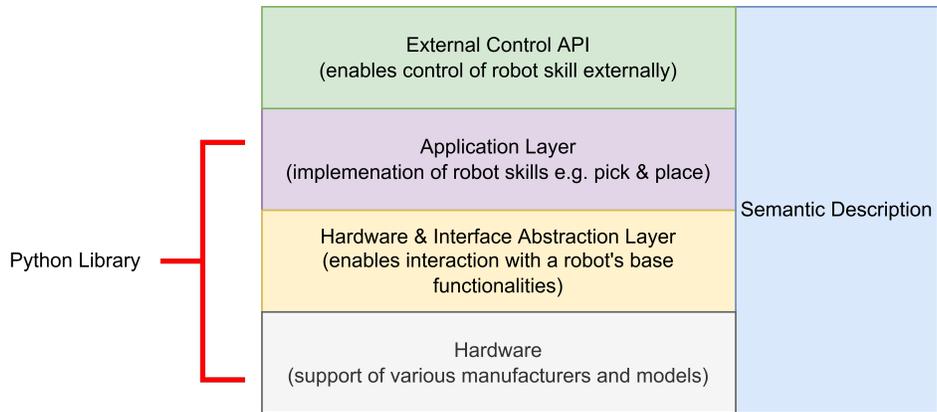


Figure 1: 所提出的软件框架由多层架构组成。最低层是支持的目标硬件。为此，只选择已经由制造商提供硬件和接口抽象的硬件。各种硬件和接口抽象被统一，以便使应用层作为 Python 库。外部控制 API 层提供基本的监控和错误处理功能。所有层都在语义描述中表示。

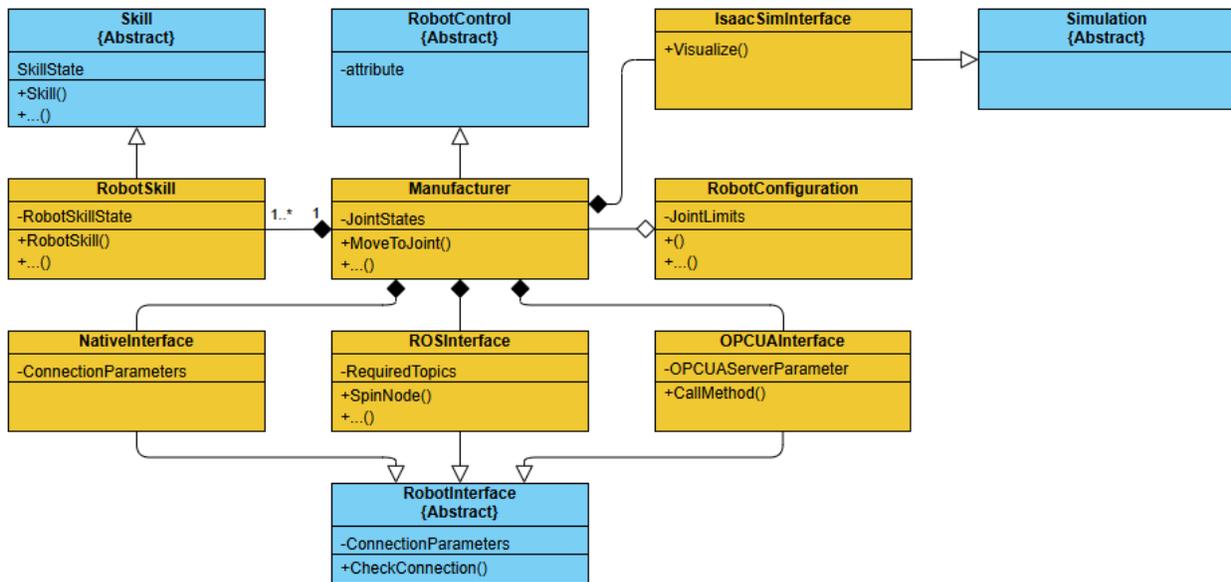


Figure 2: 类图表示软件架构，该架构使用 Python 编程语言实现。该架构是模块化的，因为库可以为每个制造商独立扩展和维护。现有硬件抽象（例如 ROS）与库的控制 API 之间的映射在制造商类中实现。机器人的功能由抽象类 RobotControl 定义的硬件无关方法实现。

我们还设置了一个抽象类用于仿真，以涵盖仿真的一些方面。目前，只有一个轻量级的 Isaac SimInterface 类使用该来启用可视化。

4.2 结果 & 讨论

Python 库的原型用于替换一个使用接近单一软件架构的软件的拣选单元控制软件，仅使用一个类通过 ROS API 进行控制。现有的软件是专门为该单元编写的，虽然后来使用了 ROS 接口来控制机器人系统。引入了代码行数 (LOC) 度量以比较我们的方法与传统方法的效率。数据仅针对单一使用案例，结果在其他使用案例中可能会有所不同。

表 1 汇总了实现箱子拾取算法所需的代码行数。旧的单片架构使用 ROS2 作为中间件来控制机器人系统，与其他硬件抽象不兼容。这导致专用代码也具有 ROS2 的功能，使算法无法用于其他接口。尽管我们的方法有更多可重用代码行，但它也允许使用其他接口，例如 OPC UA。此外，专用代码不使用任何硬件或接口特定功能，使得算法可用于其他机器人系统和接口。总之，所提出的框架将机器人系统的硬件和接口的复杂性与

Software Architecture	LoC ¹ - specialized code	LoC - reusable code
Monolithic	56	212
Our Approach	23	434

¹Lines of Code

Table 1: 该表比较了用于实现一个拣选算法的代码行数，作为一种度量标准，以突出我们的方法相比于传统单片程序设计机器人应用方法的优势。

应用层分开，以使对机器人领域知识较少的用户能够开发和部署机器人应用。用户仍需具备机器人的基本知识，特别是在安全领域。开发的 Python 库的模块化设计需要将供应商特定的实现集成为单独的包，这减少了维护工作量并提供了干净的软件结构。机器人的基本技能的实现是一个挑战，因为除了第 2 章提到的工业规范和标准外，没有任何可以使用的通用理解或定义。因此，未来的工作需要关注这一领域。此外，还需要评估实时动态场景中的性能。此外，使用该库的前提是必须设置特定接口的环境，例如 ROS2 驱动程序。

5 结论

提出的软件框架提高了机器人应用开发的效率。最实用的方面是应用算法的完整硬件和接口无关的开发。在前面提到的拣选单元使用案例中，通过简单地调整配置，所开发的代码可以在框架中重用于任何机器人制造商或型号。尽管本文的实现是原型性的，但在给定场景的运行时未检测到重大错误。这并不是说不需要严格的软件测试来确认这一方面。在未来的工作中，需要解决对模拟框架的支持。此外，机器人能力的定义需要基于详尽的文献回顾。

这项研究得到了欧洲联盟的 HORIZON 科研与创新行动计划的资助，资助协议编号为 101138782，项目为 RAASCEMAN¹ 以及德国联邦经济事务和气候行动部 (BMWK) 在 TWIN4TRUCKS² 项目 (13IK010F) 中的支持。

References

- [1] Morgan Quigley. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*, January 2009.
- [2] Steve Macenski. Robot Operating System 2: Design, Architecture, and Uses In The Wild. *Sci. Robot.*, 7(66):eabm6074, May 2022. arXiv:2211.07752 [cs].
- [3] Markus Rickert and Andre Gaschler. Robotics library: An object-oriented approach to robot applications. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 733–740, September 2017. ISSN: 2153-0866.
- [4] Rosen Diankov and James Kuffner. OpenRAVE: A Planning Architecture for Autonomous Robotics, 2008.
- [5] Toby H J Collett, Bruce A MacDonald, and Brian Gerkey. Player 2.0: Toward a Practical Robot Programming Framework. *Proceedings of the Australasian Conference on Robotics and Automation (A CRA 2005)*, December 2005.
- [6] Profibus. SRCI - Robotics.
- [7] Van Eldijk. Motion Control, April 2018.

¹<https://cordis.europa.eu/project/id/101138782>

²<https://www.twin4trucks.de/>