

Einsatz von Software-Agenten zur vereinfachten Bedienung von Robotern

Dr.-Ing. Robert Patzke
MFP GmbH, Wunstorf

AUTOMATICA 2010

Zusammenfassung

Es wird der Einsatz eines sog. Software-Agenten zur Vereinfachung der Bedienung eines Roboters erläutert. Darüber erhält der Anwender eine neue **Schnittstelle zum Roboter**, hier nach ISO 20242 ausgeführt, die sich deutlich **an seinen Aufgaben orientiert** und leicht zu beschreibende Bewegungsvorgänge ausführt. Der Software-Agent wird gezielt für eine spezifische Anwendung konfiguriert und erhält alle Informationen für die Gestaltung der Roboterbewegungen in einer leicht verständlichen, aus der Anwenderumgebung abgeleiteten, Form. Auf diesem Weg können **Bewegungsprofile** offline erstellt und für beliebige Anwendungen verwaltet werden. Das klassische Programmieren des Roboters, als Bedienung der Bewegungsschnittstelle, entfällt. Diese Aufgabe übernimmt der Software-Agent optimal für seinen Roboter. Zum Einsatz kommen Roboter, die über die **ORiN-Schnittstelle** angesprochen werden.

1 Software-Agenten

Software-Agenten sind mit hohen Erwartungen konfrontiert. Sie werden oft mit komplexen Objekten der objektorientierten Programmierung verglichen, mit zusätzlichen Ansprüchen an Fähigkeiten und Verhalten. Man erwartet, dass Software-Agenten "intelligent" sind. Eine typische Definition lautet:

Als intelligenter Software-Agent wird ein System (Softwareprogramm) bezeichnet, das für einen Benutzer bestimmte Aufgaben erledigen kann und dabei einen Grad an Intelligenz besitzt, der es befähigt, seine Aufgaben in Teilen autonom durchzuführen und mit seiner Umwelt auf sinnvolle Art und Weise zu interagieren.

2 Roboterschnittstellen

Roboter haben eine Schnittstelle, über die sie programmiert werden können. Programmieren heißt hier, den gewünschten Bewegungsablauf in den Roboter eingeben.

Jeder Roboterhersteller hat seine spezifische Programmiersprache. Versuche, eine einheitliche Robotersprache über die Standardisierung zu etablieren, sind gescheitert. In der Praxis dominieren die herstellereigenen Sprachen, wie z.B. RAPID (ABB) oder KRL (KUKA). Der Grund liegt wohl darin, dass diese Sprachen einen sehr engen Bezug zur Steuerungshardware des Roboters haben und darin Unterschiede zwischen den Herstellern deutlich werden.

Ein bemerkenswerter neuer Ansatz kommt aus Japan mit **ORiN** (Open Robot Interface for the Network, www.orin.jp/e). Hier wird eine Integrationschnittstelle definiert, die sich nicht am Roboter selbst sondern an seiner Anwendung über die objektorientierte Programmierung im Netzwerk orientiert. Da dieses Konzept prinzipiell auf alle Geräte anwendbar ist (Abb. 1), heißt ORiN inzwischen auch Open Resource Interface for the Network.

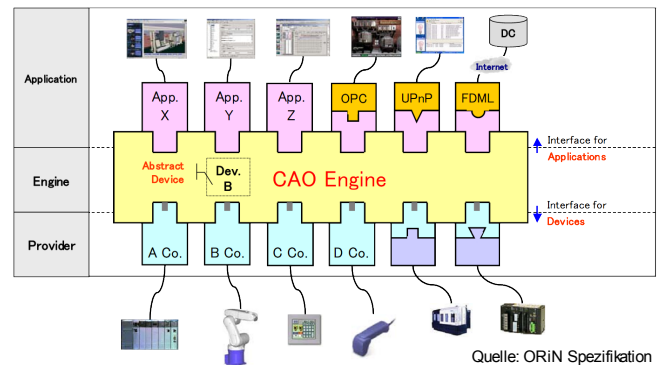


Abb. 1: Integrationskonzept von ORiN

Ein sehr ähnliches Schnittstellenkonzept wurde bereits vor vielen Jahren von der Automobilindustrie und ihren Zulieferern im ASAM (Association for Standardisation of Automation and Measurement, www.asam.net) unter dem Begriff **GDI** (Generic Device Interface) spezifiziert und veröffentlicht. Dieser Standard wird zur Zeit unter **ISO 20242** international genormt.

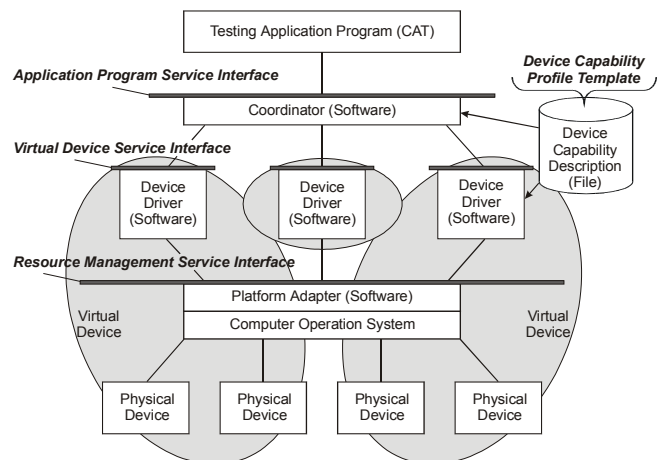


Abb. 2: Schnittstellen nach ISO 20242

Wegen der Ähnlichkeit der beiden Konzepte hat man sich bei ORiN entschlossen, eine Integration dieses Standards in ISO 20242 zu unterstützen und einen entsprechenden Anhang für den Teil 4 der Norm bereitzustellen.

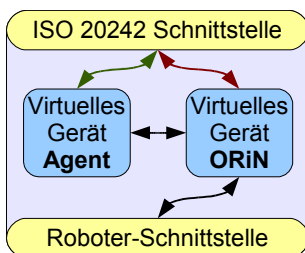
3 Neues Konzept zur Bedienung von Robotern

Nicht umsonst sind Roboterschnittstellen und Programmiersprachen sehr spezifisch und auf die Eigenschaften der Robotersteuerungen zugeschnitten. Dazu kommt eine deutliche Dynamik in der Roboterentwicklung, die weitere Ansprüche an die Roboterschnittstellen erwarten lässt.

Es bietet sich an, für einen Roboter zwei Schnittstellen anzusetzen,

- eine für die detaillierte Konfiguration und Steuerung mit dem Ziel der optimalen Nutzung der Robotersteuerung und ihrer Fähigkeiten und
- eine weitere für die möglichst einfache und übersichtliche Eingabe von Anwenderaufgaben.

Objektorientierte Schnittstellenkonzepte, wie ORiN und ISO 20242, erlauben zudem, beide Ansprüche an derselben Schnittstelle entkoppelt zu behandeln. Hier bietet sich besonders das Konzept der virtuellen Geräte nach ISO 20242 an. Es lässt sich sowohl auf das Objektmodell der klassischen Roboterschnittstelle (hier wird der sog. ORiN-Controller eingesetzt)



als auch auf den Software-Agenten zur Unterstützung der Anwendervorstellungen anwenden (Abb. 3).

Abb. 3: Getrennte virtuelle Geräte an derselben Programmierschnittstelle mit ISO 20242

Für die Anwendung des Roboters stehen an der ISO 20242 Schnittstelle nun beide funktionalen Bereiche, der für die optimale Bedienung der Steuerung und der für die Konfiguration und Beauftragung des Agenten, zur Verfügung.

In einer erweiterten Anwendung dieses Konzeptes der entkoppelten Bereiche können nun auch besonders komplizierte Aufgaben an dasselbe API (Application Programming Interface) gelegt werden. Das ist z.B. die Grundkonfiguration des Roboters oder seine Wartung. Allerdings würde man diese speziellen Funktionen vor dem typischen Anwender verbergen. Der Vorteil des API liegt hier in seiner Standardisierung, wodurch sich erhebliche Synergieeffekte durch die Verwendung von Softwaremodulen verschiedener Anbieter erreichen lassen.

4 Typische Anwenderaufgaben

Wir betrachten hier die Anwendung der sog. Offline-Programmierung. Das heißt, der Bewegungsvorgang des Roboters wird zu einer beliebigen Zeit (z.B. Planung/Konstruktion) festgelegt und dann dem Roboter (hier dem Roboter-Agenten) zur Ausführung übergeben.

Die Hauptaufgabe des Software-Agenten besteht darin, die Vorstellungen des Anwenders vom Bewegungsvorgang in die jeweilige Robotersprache umzusetzen. Dabei wird seine "Intelligenz" benötigt, um das Verhalten des Roboters aufgrund vorgegebener Ziele zu optimieren.

Naheliegende Ziele (Strategien) sind z.B. die Minimierung der Bewegungszeit oder die Minimierung des Energieverbrauches bei vorgegebener maximaler Bewegungszeit.

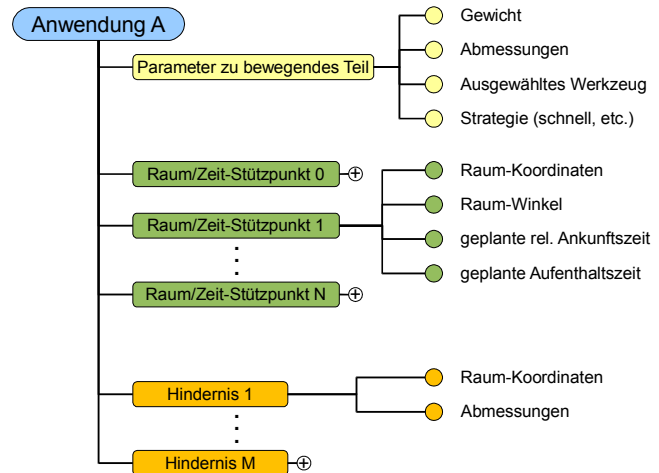


Abb. 4: Anwendungsobjekt für Roboter

Vorausgesetzt, der Software-Agent ist entsprechend leistungsfähig, so liegt es nahe, dem Anwender an der ISO 20242 Schnittstelle ein möglichst einfach zu bedienendes Objektmodell zu liefern (Abb. 4).

5 Konfiguration der Anwendung

Mit einem geeigneten Objektmodell wird, nach dem neuen Konzept mit Software-Agent, für eine geplante Roboter-Anwendung kein Programm mehr geschrieben, sondern es werden nur noch die Parameter eines konkreten Objektes festgelegt. Das geschieht im Umfeld der ISO 20242 über eine XML-Datei. Entsprechend werden die Möglichkeiten einer Anwendung mit einem XML-Schema beschrieben. Wie bei den Klassen der objektorientierten Programmierung enthält das XML-Schema strukturelle Festlegungen für einen Raum/Zeit-Stützpunkt, für ein Hindernis und für alle weiteren möglicherweise benötigten Objekte.

Die Gestaltung einer Roboteranwendung nach diesem Muster ist nicht auf den Roboter allein beschränkt, sondern erstreckt sich auch auf möglicherweise zu verwendende Hilfsgeräte. Dann kommen z.B. noch weitere Objekte wie Messaufnehmer, Trigger und Verarbeitungsverfahren dazu. Mit einem Anwendungsprogramm, wie **PACTOR**[®] von MFP, werden die Parameter einer entsprechend konfigurierten Anwendung über die ISO 20242 Schnittstelle in die Geräte übertragen (Abb. 5).

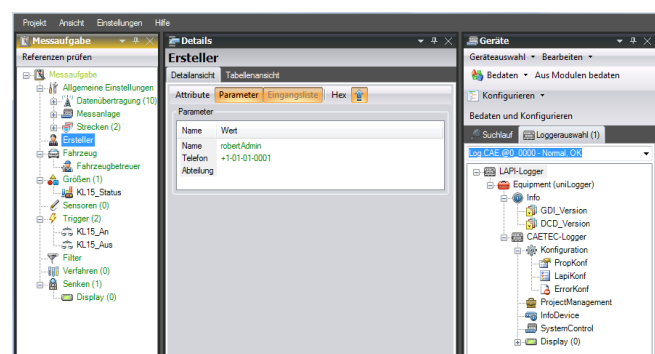


Abb. 5: Gerätekonfiguration mit PACTOR von MFP