



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Master-Seminar

Felix Kolbe

Robot Development Frameworks

Inhaltsverzeichnis

1 Einführung	3
1.1 Gliederung	4
2 Anforderungen	5
2.1 Anforderungen zur Laufzeit	6
2.2 Anforderungen zur Entwicklungszeit	6
3 Vorarbeiten, Risiken, Vorgehen	8
3.1 Schlüsse aus den Vorarbeiten	8
3.2 Mögliche Risiken	8
3.3 Weiteres Vorgehen	9
4 Auswahl verfügbarer Frameworks	10
Literaturverzeichnis	12
Abbildungsverzeichnis	13

1 Einführung

Die Welt der Roboter wird mit jedem Tag größer und komplexer. Insbesondere autonome mobile Roboter gewinnen an Zahl und Bedeutung in der Forschung; ein Teil von ihnen ist als Service-Roboter für alltägliche, menschenähnliche Hilfsaufgaben gedacht. Durch Weiterentwicklungen von Sensoren, Aktoren und weiteren Komponenten wird ein Roboter mit immer mehr Aufgaben bedacht. Da diese Aufgaben immer präziser und flexibel der Umgebung angepasst erledigt werden sollen, steigt die Komplexität der Steuerprogramme.

Um die Entwicklung der Steuer-Programme dennoch überschaubar zu halten stehen verschiedene Frameworks zur Verfügung. Mit jeweils anderen Zielsetzungen vereinfachen sie etwa die Code-Wiederverwendbarkeit, unterstützen flexible Modularisierung oder ermöglichen unkomplizierte Simulationen.

In dieser Ausarbeitung wird eine Masterarbeit geplant, welche verschiedene Frameworks untersucht, die als Softwarebasis zur Steuerung eines autonomen mobilen Service-Roboters dienen könnten. Bei dem für die Untersuchung beispielhaft betrachteten Roboter handelt es sich um die Roboterplattform „SCITOS G5“ der Firma MetraLabs¹ mit einem darauf montierten Greifarm der Firma Schunk².

Dabei sollen die vielfältigen Kriterien an ein Framework benannt und untersucht werden, sodass letztendlich ein oder mehrere einsetzbare Frameworks herausgearbeitet, beziehungsweise die Entwicklung eines eigenen Frameworks geplant werden kann.



¹MetraLabs GmbH: mobile robots – <http://www.metralabs.com>

²Schunk GmbH & Co. KG: Modular Robotics – <http://www.schunk-modular-robotics.com>

1.1 Gliederung

Zunächst werden einige grundlegende Anforderungen erläutert, welche an die Roboter-Steuerungssoftware gestellt werden. Die Anforderungen werden um weitere Kriterien ergänzt, um die Auswahl eines Frameworks zu unterstützen. Nach der Beschreibung der vorhergehenden Arbeiten werden mögliche Risiken mit ihren Wahrscheinlichkeiten und Ausweichwegen dargestellt. Im Anschluss folgt die Schilderung des weiteren Vorgehens. Abschließend werden mögliche Framework-Alternativen mit einigen Besonderheiten beleuchtet.

2 Anforderungen

Folgende elementare Anforderungen werden an die Steuersoftware des untersuchten Scitos@HAW-Service-Roboters gestellt:

Echtzeitfähig

Wenn sich Roboterarm oder -plattform bewegen, müssen Sensorwerte ausreichend schnell verarbeitet werden, um plötzlich erscheinenden Hindernissen rechtzeitig ausweichen zu können. Mit der gewünschten Bewegungsgeschwindigkeit steigt die Echtzeit-Anforderung.

Nebenläufigkeit

Die Ausführung mehrerer Prozesse oder Threads muss gut parallelisierbar sein, damit wichtige zeitkritische Regelalgorithmen und unwichtige Datenverarbeitungsprozesse unabhängig voneinander ablaufen können.

Zentrale Datenverteilung

Zwischen den Prozessen mit ihren Teilaufgaben müssen verschiedenste Informationen und Befehle ausgetauscht werden. Um Inkonsistenzen durch verworrene Informationskanäle zu verhindern, ist eine zentrale und globale Datenverteilung wünschenswert.

Vernetzung

Der Roboter soll in einer intelligenten Umgebung¹ eingesetzt werden, sodass ihm zusätzliche Informationen von außen zur Verfügung stehen. Falls die Leistung des Roboter-Steuercomputers für komplexere Programme (bspw. zur Bildverarbeitung) nicht mehr ausreicht, wäre der Einbau weiterer Hardware notwendig. Beide Szenarien erfordern eine performante Vernetzung.

Sensoreinbindung

Der bestehende Roboter soll zukünftig um weitere Sensoren oder andere Komponenten erweitert werden. Dies bedeutet, dass alle gängigen oder besonderen Schnittstellen mit akzeptablem Programmieraufwand von der Steuersoftware ansprechbar sein müssen. Eine Besonderheit stellt die MetraLabs-Bibliothek des Roboters dar: Sie erst ermöglicht den Zugriff auf die üblicherweise im Scitos verbauten Sensoren und muss daher auf geeignete Weise in die Steuersoftware eingebunden werden können.

¹Forschungswohnung Living Place der HAW Hamburg: <http://livingplace.informatik.haw-hamburg.de>

Diese Anforderungen werden im Folgenden erweitert und in die beiden Bereiche Laufzeit und Entwicklungszeit unterteilt. Diese sollen später zur Unterscheidung und Bewertung verfügbarer Frameworks dienen. Es ist gut möglich, dass diese Gruppierung in der Masterarbeit verfeinert wird.

2.1 Anforderungen zur Laufzeit

Diese Anforderungen betreffen die Performance des Systems, insbesondere bezüglich Reaktionszeit und Datendurchsatz. Sie bilden die wichtigste Gruppe von Anforderungen, da ohne ein funktional lauffähiges System die Nebenaspekte unbedeutend sind.

- Flexibilität bzgl. HW-Schnittstellen
- Flexibilität bzgl. Modularisierung
- Zentrale Datenhaltung, Koordinierter Informationsfluss
- Skalierbarkeit, Vernetzbarkeit
- Performance, Nebenläufigkeit
- Priorisierbarkeit & Echtzeitfähigkeit

2.2 Anforderungen zur Entwicklungszeit

Die zweite Gruppe von Anforderungen betrifft nützliche Funktionen, die dem Programmierer zur Entwicklungszeit zur Verfügung stehen und einen geordneten Entwicklungsprozess fördern.

- Strukturunterschiede, Entscheidungshierarchie
- Unterstützung von Simulationen / virtueller Hardware
- Unterstützung von Modellen und Debugging
- Bereitstellung von Bibliotheken vorgefertigter Funktionen
- Bereitstellung / Unterstützung von Entwicklungswerkzeugen
- Paket- und Versionskontrolle
- Vorausgesetztes OS + Grafiksystem
- Programmiersprachenunabhängigkeit

- Einarbeitungszeit und Lernaufwand bei z.B. unbekanntem Skriptsprachen
- Projekt-Aktualität, Community-Aktivität

3 Vorarbeiten, Risiken, Vorgehen

3.1 Schlüsse aus den Vorarbeiten

In der Projektarbeit ([Kolbe, 2010a](#)) wurde der Scitos G5 bereits in Betrieb genommen und programmiert. Der Aufbau und die Schnittstellen der ML-Bibliothek [G5 Software Guide \(2010\)](#) sind daher hinreichend bekannt, ebenso wie die Anbindung der Hardware-Komponenten.

In der Theoriearbeit ([Kolbe, 2010b](#)) wurde recherchiert, welche Frameworks zur Verfügung stehen und welche von diesen in vergleichbaren Roboterprojekten verwendet werden.

3.2 Mögliche Risiken

Es bestehen einige Risiken, die zu einem teilweisen oder vollständigen Misslingen der Framework-Integration führen können.

Wie in den [Anforderungen](#) beschrieben, ist die Einbindung in die bestehende Umgebung aus Hardware und MetraLabs-Umgebung notwendig. Wenn das erwählte Framework trotz gegenteiliger Annahmen dazu nicht kompatibel sein sollte, wäre ein weiterer Wechsel, eine Anpassung des Frameworks oder Support seitens MetraLabs erforderlich. Beide Alternativen wären mit einem deutlichen Zeitaufwand verbunden.

Ein globales Risiko stellt die Unabsehbarkeit des zukünftigen Performancebedarfs dar. Die Leistungsfähigkeit des Frameworks, sowie das zu erwartende Datenaufkommen und die Echtzeitanforderungen lassen sich kaum vorraussehen und nur schwer in Zahlen festlegen. Um hierbei genügend Auswege z.B. in Form von leistungsstärkerer oder zusätzlicher Rechenhardware zu ermöglichen, ist unbedingt auf eine flexible Struktur zu achten.

Ein langfristiges Risiko besteht in der Aktivität der Entwickler des gewählten Frameworks. Schläft die Entwicklung ein, versiegt der Projektsupport. Umso fataler wäre dies bei Projekten mit unfreiem Quelltext.

3.3 Weiteres Vorgehen

Die Anforderungen des Roboterprojektes Scitos@HAW werden genauer analysiert und soweit möglich in messbare Zahlen gefasst. Zu den bereits genannten [Anforderungen](#) können dabei weitere hinzukommen. Dabei ist es wichtig, neben den aktuellen Spezifikationen alle wahrscheinlichen zukünftigen Systemerweiterungen miteinzubeziehen. Auch sind Kenntnisse über die Entwicklungs- und Einsatzumgebung festzuhalten und zu berücksichtigen.

Dabei ist eine Gewichtung der Anforderungen sinnvoll. Hierbei sind zuvor beschriebene Risikokriterien besonders zu beachten. Anhand dieser Anforderungen wird die Eignung sämtlicher verfügbarer Frameworks untersucht. Dabei werden folgende Varianten als Untersuchungsergebnis erwartet:

Kein verfügbares Framework erfüllt die Anforderungen

Sollte kein verfügbares Framework den gestellten Anforderungen genügen, wird die Eigenentwicklung eines zum Scitos@HAW passenden Frameworks gestartet. Dabei muss ein Konzept erarbeitet werden, welches alle Anforderungen mit ihrer jeweiligen Priorität berücksichtigt.

Ein verfügbares Framework erfüllt die Anforderungen

Im vorerst einfachsten Fall erfüllt lediglich ein Framework alle Kriterien. Um die Eignung des Frameworks zu beweisen, wird dieses auf dem Scitos@HAW-System installiert und mit realistischen Programmen verwendet.

Mehrere verfügbare Framework erfüllen die Anforderungen

Falls mehrere Frameworks den Kriterien genügen und sich diese nennenswert unterscheiden, steht eine weitere Untersuchung an. Dabei soll zum einen geprüft werden, ob die Frameworks sich tatsächlich mit dem Scitos verwenden lassen, zum anderen sollen über realistische Testprogramme Leistungsunterschiede herauskristalisiert werden.

4 Auswahl verfügbarer Frameworks

In diesem Abschnitt werden vier der möglichen Frameworks betrachtet und einige Besonderheiten herausgestellt.

MetraLabs-Blackboard-Architektur

Die für den Scitos erforderliche ML-Bibliothek beinhaltet bereits Strukturen, um mehrere Prozesse mit ihren Daten zu steuern. Dazu gehört an zentraler Stelle das Blackboard, welches den öffentlichen Aushang aller Daten von Sensoren und Prozessen darstellt. Hierüber besteht die Möglichkeit, sämtliche Daten aufzuzeichnen, sowie Aufzeichnungen abzuspielen, um Simulationen durchführen oder Situationen reproduzieren zu können. Abbildung 4.1 zeigt die Aufteilung der ML-Bibliothek mit ihren optionalen Komponenten.

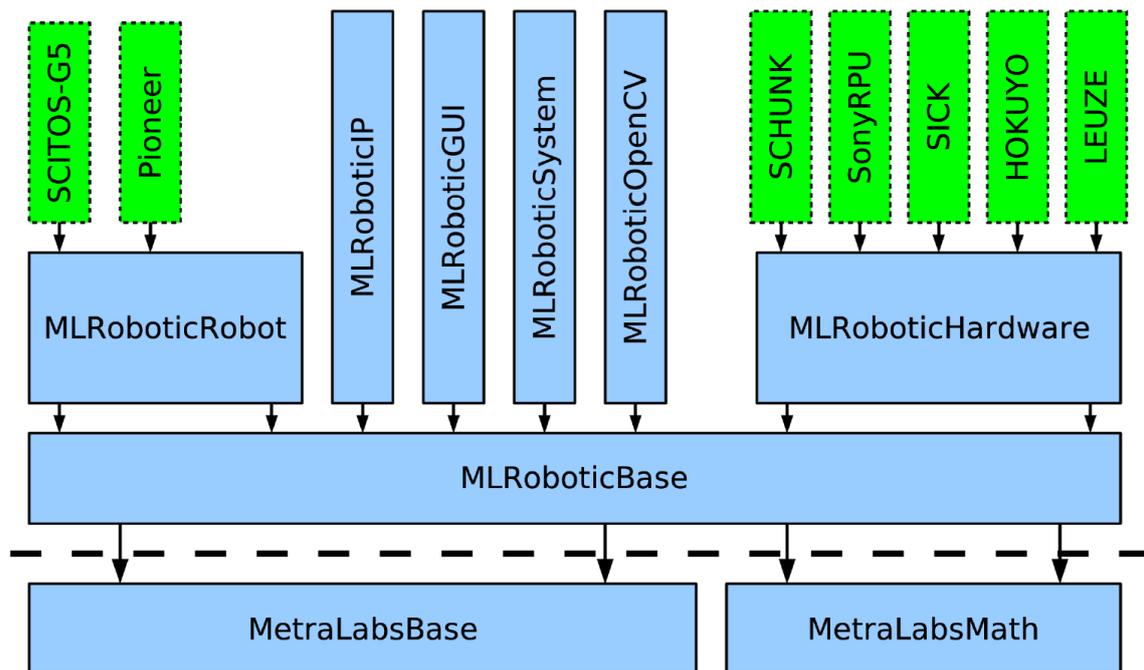


Abbildung 4.1: Anordnung der MetraLabs-Bibliotheken

BRICS (<http://www.best-of-robotics.org>)

BRICS ist ein Framework, welches besonderen Wert auf den Entwicklungsprozess

legt, sodass die Entwicklung eines Roboters, wie bereits in der Softwareentwicklung üblich, auf einem strukturierten Prozess anstatt einer findigen Tüftelei basiert (vgl. [BRICS](#)). Dazu sollen nicht nur wie üblich Funktionsbibliotheken und Softwareinfrastruktur bereitgestellt werden, sondern auch Werkzeuge, mit denen formale Modelle erstellt, validiert und wiederverwendet werden können, das Blackboard befindet sich in MLRoboticBase.

FAUSTCore (<http://www.informatik.haw-hamburg.de/faust.html>)

Das Projekt FAUST der HAW Hamburg entwickelt Fahrerassistenz- und Autonome Systeme auf verschiedenen kleineren Fahrzeugenplattformen. Der FAUSTCore stellt die zentrale Softwarekomponente dar, über die verschiedene Regelprozesse zusammengeschlossen und Konfigurationen verwaltet werden. Ein Bonus läge hierbei in der Hausinternen Zusammenarbeit.

Player/Stage (<http://playerstage.sourceforge.net>)

Das Player Project besteht hauptsächlich aus einem Netzwerkserver („Player“), der auf dem Robotersystem läuft und die Sensoren und Aktoren über das Netzwerk ansprechbar macht. Da die Netzwerk-Kommunikation die einzige Schnittstelle darstellt, kann die Steuersoftware in einer beliebigen Sprache auf beliebiger Plattform geschrieben werden und auch aus mehreren Rechnern bestehen (vgl. [Player](#)). „Stage“ ist die zugehörige Simulationsplattform.

Literaturverzeichnis

- [BRICS] BRICS, BEST PRACTICE IN ROBOTICS: *BRICS in a nutshell*. – URL <http://www.best-of-robotics.org/en/brics-in-a-nutshell.html>. – Abruf: 2011-02-26
- [G5 Software Guide 2010] METRALABS (Hrsg.): *SCITOS G5 Software Guide*. 2010. – URL http://www.metralabs-service.com/downloads/SCITOS/doc/SCITOS_G5-Software_Guide_v1.17.pdf. – Abruf: 2010-09-05
- [Kolbe 2010a] KOLBE, Felix: *Inbetriebnahme und Programmierung der Scitos G5-Roboterplattform : Projektausarbeitung*, Hamburg: Hochschule für Angewandte Wissenschaften, Dep. Informatik, Ausarbeitung, September 2010
- [Kolbe 2010b] KOLBE, Felix: *Indoor-Navigation eines autonomen mobilen Service-Roboters : Vorstellung vergleichbarer Projekte*, Hamburg: Hochschule für Angewandte Wissenschaften, Dep. Informatik, Ausarbeitung, August 2010
- [Player] THE PLAYER PROJECT: *Player : Cross-platform robot device interface & server*. – URL <http://playerstage.sourceforge.net/index.php?src=player>. – Abruf: 2011-02-26

Abbildungsverzeichnis

4.1 Anordnung der MetraLabs-Bibliotheken	10
--	----