

Hochschule Darmstadt
- Fachbereich Informatik -

Untersuchung der Offenen Schnittstellen des Ur5 Roboters anhand
eines Anwendungsbeispiels

Abschlussarbeit zur Erlangung des akademischen Grades Bachelor of Science
(B.Sc.)

vorgelegt von

Andreas Collmann

Referent: Prof. Dr. Horsch
Korreferent: Prof. Dr. Akelbein

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, den 28.03.2014

Abstract

Um die Vorteile des Kollaborativen Arbeitens von Menschen und Robotern anzuwenden, wird im Zuge dieser Arbeit der für die Kollaboration zugelassene Roboter UR5 der Firma Universal Robots untersucht. Es werden die Möglichkeiten diesen Roboter zu programmieren untersucht. Die Untersuchung erfolgt aufgrund einiger Kriterien, die auf den Einsatz mit Roboter Mensch Kollaboration zielen. Die Schnittstellen des UR5 Roboters werden untersucht und dokumentiert. Die Ergebnisse dieser Arbeit, fassen eine Entscheidungsfindung zusammen, welche Schnittstelle zu welchem Anwendungsfall am besten zu wählen ist.

Inhaltsverzeichnis

1. Einleitung	10
1.1. Fachliche Umgebung	10
1.2. Motivation und Ziel des Projektes	10
1.3. Aufgabenstellung	11
1.4. Einordnung in die Themenfelder der Informatik	11
2. Grundlagen	12
2.1. Roboter-Mensch-Kollaboration	12
2.1.1. Richtlinien	12
2.2. Ur5 Roboter	13
2.2.1. Eigenschaften	13
2.2.2. System Aktualisieren	13
2.2.3. Datensicherung	14
2.3. Übersicht der Programmierschnittstellen	14
2.4. Kriterien für die Bewertung der Schnittstellen	15
2.5. URControl	16
2.5.1. Konfiguration des URControllers	16
2.5.2. Echtzeit Schnittstelle	16
2.5.3. Secondary und Primary Schnittstelle	17
2.5.4. Polyscope	18
2.6. C-API	18
2.6.1. Kontrollstruktur	18
2.6.2. PTP Verfahren	19
2.6.3. Linear Verfahren	19
3. Konzept	20
3.1. Anwendungsbeispiel	20

3.2.	Speichern der Anwendungsdaten	20
3.2.1.	Speichern über Polyscope und URScript	20
3.2.2.	Speichern über Eigene API	21
4.	Realisierung	22
4.1.	TCP Server mit Datenbank zum dauerhaften speichern der Daten . .	22
4.2.	C-API	22
4.2.1.	gelöste Aufgabe	22
4.2.2.	Aufgetretene Probleme	23
4.3.	Polyscope	23
4.3.1.	Programmierung	23
4.3.2.	Benutzer Interaktion	23
4.3.3.	Test und Fehlersuche im Programm	24
4.3.4.	Aufwand der Programmierung	24
4.4.	URScript	24
4.4.1.	Laden des Scripts auf den Controller	25
4.4.2.	Programmierung	25
4.4.3.	Test und Fehlersuche im Programm	25
4.4.4.	Benutzer Interaktion	26
4.4.5.	Aufwand der Programmierung	26
4.5.	Anwendung mit Eigener API	26
4.5.1.	Adapter zur Secondary Schnittstelle	26
4.5.2.	Programmierung mit Adapter	27
4.5.3.	Benutzer Interaktion	27
4.5.4.	Test und Fehlersuche im Programm	27
4.5.5.	Aufwand der Programmierung	27
5.	Ergebnis	28
5.1.	Vergleich der Schnittstellen	28
5.2.	Nicht erreichte Ziele	28
5.2.1.	C-API	29
6.	Fazit	30
6.1.	Ausblick	30

6.2. Zusammenfassung	30
6.2.1. C-API	30
6.2.2. UR-Script	31
6.2.3. Polyscope	31
6.2.4. Eigene API	31
6.2.5. Grobe Empfehlung, welche Schnittstelle, bei welchem An- wendungsfall verwendet werden soll	32
A. Literaturverzeichnis	33
B. Glossar	34
C. Quellcode	35
C.1. Speichern der daten über TCP in der Datenbank	35

Abbildungsverzeichnis

2.1. Schichten der Software Schnittstellen	15
2.2. Schema des Datenpackets gesendet von der Secondary Schnittstelle	17

Tabellenverzeichnis

Listings

2.1. Pfade Der Ur5 Relevanten Dateien	14
---	----

1. Einleitung

1.1. Fachliche Umgebung

Hauptaugenmerk dieser Arbeit ist es die Möglichkeiten der Roboter-Mensch-Kollaboration in der Industrie, Medizin, Schule mit diesem Roboter zu prüfen. Inwiefern der Mensch mit einem heutigen Roboter mit entsprechenden Richtlinien programmiert werden kann um in den entsprechenden Feldern mit dem Menschen zusammen zu arbeiten.

1.2. Motivation und Ziel des Projektes

In der Industrie werden Roboter in den Fertigungsanlagen eingesetzt. Dies geschieht meist in Koordination mit anderen Robotern. In der Nähe dieser Roboter, darf sich kein Mensch aufhalten, die Roboter sind umhaust, sprich in einem speziellen Bereich abgesichert, damit keine Unfälle passieren können. Auf diese Weise kann man sehr effizient über automatisierte Fließbandstraßen Produkte herstellen.

Wenn jedoch eine sehr filigranere Arbeit gefragt ist, muss das Werkstück von einem Menschen bearbeitet werden, da der Mensch wesentlich bessere Fähigkeiten hat, auf Probleme zu reagieren oder Korrekturen vorzunehmen. In diesem Fall wird die Fließbandstraße unterbrochen. Das Produkt muss aus dem umhausten Bereich gebracht werden, wo es von einem Menschen bearbeitet werden kann.

Für die Produktion wäre es viel sinnvoller und zeitsparender, wenn Roboter für den Menschen so sicher sind, dass keine Trennung zwischen Mensch und Robotern existiert.

In dem Bereich Pflege und der Medizin, müssen oft hebearbeiten ausgeführt werden. Dies führt dazu, dass die Menschen in solchen Berufen im späteren Alltag mit Rückenproblemen oder ähnlichen leiden leben müssen. Roboter, die eingesetzt werden um diese Lasten abzunehmen, würde die Arbeit erleichtern und Verletzungen

vorbeugen.

Es soll untersucht werden inwiefern die Zusammenarbeit von Robotern und Menschen mit einem Roboter der die Sicherheitsauflagen erfüllt, realisiert werden kann.

1.3. Aufgabenstellung

Es soll ein Anwendungsprogramm für alle möglichen Programmierschnittstellen für den Ur5 Roboter von Universal Robots entwickelt werden. Dieses Anwendungsprogramm soll so ausgelegt sein, dass es als eine Beispielanwendung einer Roboter-Mensch Kollaboration ist. Diese verschiedenen Programme werden miteinander verglichen. Es soll eine Entscheidungshilfe gegeben werden, für welchen Anwendungsfall, welche Schnittstelle am besten geeignet ist.

Die Programmierschnittstellen sollen möglichst gut dokumentiert werden.

1.4. Einordnung in die Themenfelder der Informatik

Die Schnittstellen werden mit den Standard Programmiersprachen C/C++ und Python programmiert. Hinzu kommt noch von der eigens von Universal Robots entwickelte URScript Sprache. Da versucht wird den Roboter von einem anderen Rechner zu steuern, wird auch Netzwerkprogrammierung benötigt. Es muss ein kleines Protokoll entwickelt werden, mit dem der Roboter kommunizieren kann um Anwenderdaten zu speichern.

2. Grundlagen

2.1. Roboter-Mensch-Kollaboration

Man unterscheidet die Arbeiten mit einem Roboter unter mehrere Arten. Roboter die mit anderen Robotern gleichzeitig arbeiten nennt man Kooperation zwischen Robotern. Der Mensch ist in diesem Arbeitsumfeld nicht dabei und kann nur von außen einfluss nehmen.

Als nächstes gibt es die Kollaboration zwischen dem Roboter und dem Mensch. Hier wird auch eine Unterteilung vorgenommen die unterschiedliche Richtlinien erfordern.

- Sicherheit, wenn der Mensch den Kollaborationsraum betritt.
- Dauerhafte Überwachung des abstands zwischen Mensch und Roboter, der mit reduzierter geschwindigkeit arbeitet.
- Verminderte Geschwindigkeit. Führung des Roboters durch den Mensch. Sensoren erfassen die kräfte, die vom Menschen ausgeführt werden und übertragen sie auf den Roboter.
- Beschränkung der im Roboter ausgeführten Energie. Überwachung des Roboters auf Kollision und sofortigem Stop

2.1.1. Richtlinien

In so gut wie allen fällen sind Roboter in der Industrie in einem extra abgesicherten Bereich umzäunt, damit kein Arbeiter sich verletzen kann. Diese Roboter sind umhaust. Es ist nicht möglich in einem gemeinsamen Arbeitsbereich zu kollaborieren. Damit Menschen im Arbeitsbereich vom Robotern Arbeiten dürfen müssen

diese Roboter bestimmte Sicherheitsrichtlinien entsprechen. Der Roboter darf unter keinen Umständen eine Lebensbedrohliche Gefahr darstellen. Die Norm ISO 10218

2.2. Ur5 Roboter

Die Dänische Firma Universal Robots hat den leichten Ur5 und mittelgroßen Ur10 Roboter hergestellt mit erfüllbaren Normen um mit diesem Roboter zu kollaborieren. Man kann sich im laufenden Betrieb in der Nähe aufhalten um Wegpunkte zu Teachen oder auch gleichzeitig an einem Werkstück zu arbeiten.

2.2.1. Eigenschaften

Der Roboter besitzt 6 Gelenke die ihm ermöglichen einen 360° Arbeitsbereich mit einem Radius von ca 85cm zu ermöglichen. Gesteuert wird er von einem Linux Rechner, der sich in der Nähe befindet. Die Festplatte für das System ist eine Speicherkarte, die leicht ausgetauscht werden kann.

Um den Rechner anzusprechen existiert bei Lieferung ein Touch Tablet, das für das Linux System den Visuellen Output gibt. bei Start wird auch automatisch die Software für den Roboter gestartet. Die Software nennt sich Polyscope und wurde in Java geschrieben. Diese Software verbindet sich per TCP/IP auf den URController. Ein Server Programm das die Schnittstelle von dem Linux System zu dem Roboter Controller auf dem Rechner herstellt.

Die Polyscope Software läuft im normalen Modus und den Administrativen Modus. Der Normale Modus ermöglicht es Programme zu erstellen, laufen zu lassen und Grundeinstellungen vorzunehmen. Außerdem kann ein Software Update der Polyscope Software gemacht werden.

2.2.2. System Aktualisieren

Zwei Arten von Updates sind hier zu unterscheiden. Zum einen kann das Linux System geupdatet werden. Auf normalem wege über den Packetmanager des Systems, oder wenn man das neuste Image von Universal Robots runterläd und dann das System neu aufspielt. Hier ist jedoch zu beachten, dass dabei alle Daten verloren

gegangen werden. Deshalb sollte eine Datensicherung vorgenommen werden. Wie dies geschieht wird im darauf folgenden Unterkapitel beschrieben(2.2.3).

Updates für den Roboter müssen allerdings manuell gemacht werden. Hierfür müssen die aktuellen Updates von der Homepage von Universal Robots heruntergeladen werden. Die Update-Datei muss mit der Dateiendung .urup auf einen USB Stick mit einem FAT32 Dateisystem abgelegt werden.

Nachdem der USB Stick an das Linux System angeschlossen ist, kann von der Polyscope Software das Update ausgeführt werden. Einstellungen->Updates.

Im Administrativen Modus können nach dem Update die Firmware's der einzelnen Gelenkcontroller geupdatet werden. Die werden im Update mitgeliefert. Die einzelnen Schritte sind in den Dokumentationen beiliegend auf der CD zu finden.

2.2.3. Datensicherung

Die Daten des Roboters sind abgelegt in root verzeichniss unter

```
1 /root/.urcontrol      #Konfigurationsdateien des Ur5Roboters
2 /programs             #alle geschriebenen Programme unter Polyscope
```

Listing 2.1: Pfade Der Ur5 Relevanten Dateien

Es ist möglich die Dateien per USB Stick zu sichern oder über Programme wie "SCP" über das Netzwerk zu Kopieren.

2.3. Übersicht der Programmierschnittstellen

Der Ur5 Roboter kann auf drei Ebenen angesprochen werden.

- Polyscope
- URscript
- C-API

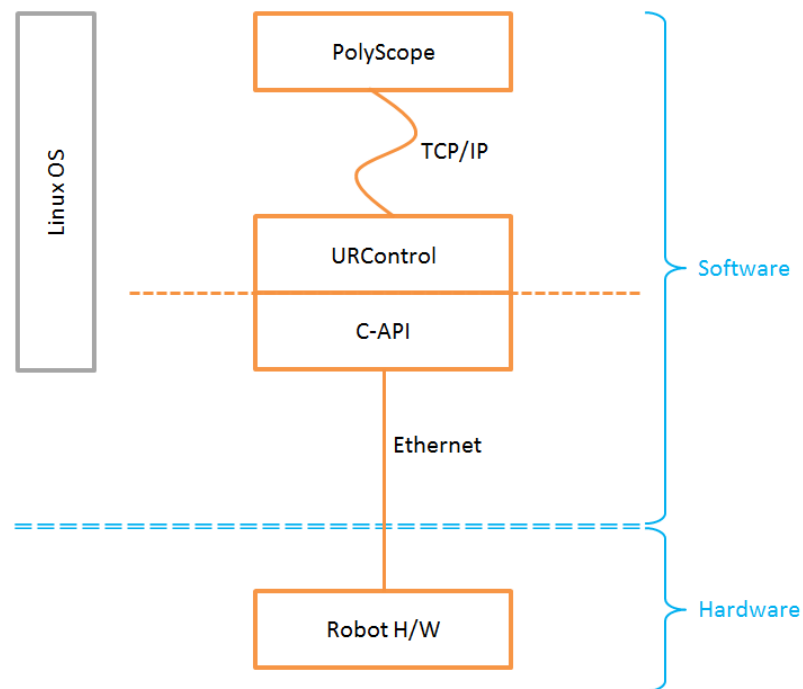


Abbildung 2.1.: Übersicht über die Schichten der bestehenden Software Schnittstellen des Ur5 Roboters

2.4. Kriterien für die Bewertung der Schnittstellen

Die Schnittstellen werden wie folgt bewertet:

- Programmierbarkeit
- Interaktion mit Programm,
- Möglichkeit zu Debuggen und Testbarkeit
- Aufwendung

Wie schwer ist es ein Programm für die einzelnen Schnittstellen zu entwickeln. Kann der Mensch das Programm intuitiv bedienen? Wichtig hierbei ist, dass der Mensch mit dem Roboter kommunizieren kann. Dies geschieht am besten, wenn der Mensch nicht Kryptisch was eingeben muss. Der Mensch braucht Anwenderfreundliche Programme.

Beim Entwickeln von Programmen ist es wichtig, dass der Entwickler Fehler im Programm entdeckt um diese schnell zu beheben. Je Größer und Komplexer das Programm wird, desto schwieriger wird es Fehler zu entdecken.

2.5. URControl

Der URController eine Server Anwendung die auf dem Rechner des Roboters läuft. Dieser Controller dient als Schnittstelle von der Roboter Hardware und Software die den Roboter ansteuern wollen.

2.5.1. Konfiguration des URControllers

Den URController kann man bevor er gestartet wird in einer Konfigurationsdatei konfigurieren. Hier werden wichtige Einstellungen vorgenommen, die zu den jeweiligen Modellen der Ur5 oder Ur10 Serie gehören. Folgend ist ein ausschnitt der Konfigurationsdatei zu sehen

```
1 [Config]
2 # masterboard_version, 0 = Zero-series, 1 = One-series,
3 # 3 = Pause function enabled, 4 = first cb2 version, 5 = ur10 support added
4 masterboard_version = 4
5 dump_bytecode_on_exception = 1
6
7 [Hardware]
8 controller_box_type = 2 # 1=CB1, 2=CB2UR5, 3=CB2UR10
9 robot_type = 1 # 1=UR5, 2=UR10
10 robot_sub_type = 1
```

2.5.2. Echtzeit Schnittstelle

Die Echtzeit Schnittstelle ist eine TCP Schnittstelle, die im 125Hz Takt Nachrichten an die Clients sendet. Diese Schnittstelle empfängt keine Daten von den Clients. Diese Nachrichten müssen von den Clients analysiert und zerlegt werden. Die Daten werden in einer bestimmten form gesendet

TODO !! listing echtzeit schnittstelle

Die Client Anwendung muss nun dieses Packet parsen(Parser: Informationen zerlegen und entsprechend interpretieren.) Wie Die einzelnen Packete aussehen sind

im Anhang mitgeliefert Für die Programmiersprache C wurde ein ein Parser dafür geschrieben.

2.5.3. Secondary und Primary Schnittstelle

Das Secondary Interface ist eine TCP Schnittstelle, die in einem 60Hz Takt Nachrichten über den Roboter an Verbundene Rechner sendet. Die Nachrichten beinhalten Informationen wie z.B. den Roboter Status, die Positionen der einzelnen Joints. Die volle Beschreibung welche Informationen gesendet werden ist im anhang zu finden.

Zusätzlich, kann die Secondary Schnittstelle befehle von Verbundenen Rechnern empfangen. Diese Befehle können URScript befehle sein. Ein ganzes Programm aus URScript befehlen oder spezielle zugelassene Befehle die den Roboter Status verändern.

4 bytes (int)	Length of overall package
1 byte (uchar)	Robot MessageType
4 bytes (int)	Length of Sub-Package
1 byte (uchar)	Package-Type
n bytes	Content...
4 bytes	Length of Sub-Package
1 byte	Package-Type
n bytes	Content...
	...

Abbildung 2.2.: Grobe Darstellung wie das Nachrichten Packet gesendet von der Secondary/Primary Schnittstelle.

2.5.4. Polyscope

Polyscope ist eine Anwendung die auf dem Roboter-Rechner läuft. Die Anwendung verbindet sich per TCP/IP auf den UR Controller und sendet URScript befehler an den Roboter um diesen zu steuern. Diese Anwendung wird auf dem Tablet angezeigt. Hierrüber kann man per Touch eingabe ein neues .URP Programm erstellen. Dieses Programm wird zur Laufzeit in ein .script umgewandelt.

2.6. C-API

Die C-API ist von dem Hersteller UR eine zur Verfügung gestellte C Library mit einer Header Datei, die etwaige Funktionen der Library erklärt. Die Header Datei enthält nicht alle Funktionen, somit sind nicht alle zugänglich. Die C-API erlaubt es einen eigenen Controller für den Roboter zu entwickeln. Der für den Roboter zur Verfügung gestellte Controller mit der Polyscope Software können aber nicht gleichzeitig laufen. Es schließen sich also die URScriptsprache und ein eigener Controller zunächst aus. Es könnte ein eigener Controller gebaut werden der die Befehle in URScript selbst interpretiert und diese wie bei dem URController ausführt. So könnte man die vorhandene Sprache nehmen und diese sogar erweitern.

2.6.1. Kontrollstruktur

Die C-API ermöglicht es eine Verbindung zum Roboter zu öffnen und über eigene Funktionen Befehle abzuschicken. Dies erfolgt in einem streng festgelegten Muster. Dieses Muster ist in Abbildung ... TODO !! abilbundsref zu sehen. die Funktion `robotinterface_read_state_blocking()` startet den Bereich in dem Datenabfragen an den Roboter gestellt werden können. Daten wie zb. Temperatur der Motoren, der Stand der Gelenke, die Geschwindigkeit der Gelenke, etc. in der Dokumentation beiliegend zu dieser Arbeit sind alle Daten noch einmal aufgelistet. Nachdem die Daten abgefragt werden, kann mit C-API Funktionen Position, Geschwindigkeit und Beschleunigungswerte übermittelt werden, die der Roboter durch seinen Regler auszuführen versucht.

Es können jedoch keine Wegpunkte festgelegt werden, die dann automatisch vom Roboter angefahren werden. Dies muss der Entwickler selbst berechnen. Es gibt mehrere Verfahren, in dieser Arbeit sind PTP-Verfahren(siehe Kapitel [2.6.2](#)) und

Linear Verfahren(siehe Kapitel 2.6.3) getestet worden. In der beiliegenden Dokumentation ist aufgeführt wie man dies möglicherweise berwerkstelligen könnte.

Zum Abschluss wird die Function `robotinterface_send()` aufgerufen die dafür sorgt, dass der Acht millisekundentakt eingehalten wird und die Befehle an den Roboter weiterleitet. Falls die Acht Millisekunden überschritten werden, wird der Roboter in einen Sicherheitsmodus gesetzt und der Roboter wird angehalten.

Wenn so etwas im UR-Kontroller passiert, kann der Anwender diese wieder abschalten wenn alles in Ordnung ist. Dies muss mit der C-API selbst geschrieben werden. Die API liefert hierfür auch funktionen. Das die richtigen Richtlinien aber auch eingehalten werden, muss von dem wechsel des Sicherheitsmodus in den normalen Modus eine Benutzerabfrage verlangt werden.

2.6.2. PTP Verfahren

Um den Roboter bestimmten Wegpunkten abfahren zu lassen, muss man die Bewegungsprofile selbst berechnen und über die C-API an den Roboter im 125Hz Takt übergeben. Das PTP Verfahren setzt dabei voraus das die einzelnen Positionen der Gelenke bekannt sind. Der Wert ist angegeben in radiant. Die Zielposition

2.6.3. Linear Verfahren

Das Lineare Verfahren bedeutet eine Bewegung des Roboters von dem TCP Punkt aus. Sie verfährt Linear, bedeutet das die Orientierung des TCP Punktes sich nicht ändert. Da in diesem Verfahren die Berechnung der Position des TCP Punktes verwendet wird, ist es nötig die Position des TCP im Raum zu kennen um eine Strecke zu einem Ziel Punkt abfahren zu können. Der Roboter kann aber nur Positionen die der Sechs Gelenke verarbeiten. Es muss also eine umrechnung stattfinden. Diese umrechnung wird Inverse Kinematic genannt. Die Berechnung für die Inverse Kinematic ist von einem anderen Projekt entnommen worden.

3. Konzept

3.1. Anwendungsbeispiel

Das Anwendungsbeispiel ist ein Kinderspiel. Dieses Spiel soll die motorischen Fähigkeiten bei Kindern verbessern. Gegeben ist eine Kugel mit Löchern aus verschiedenen Formen (Kreis, Oval, Viereck, Trapez, etc.). Zu diesen Formen existieren die entsprechenden Klötzchen, die entsprechend groß sind und die Form der Löcher besitzen. Die Aufgabe des Spielers ist es, alle Klötzchen in die entsprechende Form zu drücken, bis alle in der Kugel sind.

Die Kugel wird an den Kopf des Roboterarms befestigt. Es soll eine Anwendung entwickelt werden, die für den entsprechenden Spieler die Höhe des Roboters einstellt. Der Spieler soll die Möglichkeit haben, diese Startposition zu verstellen und für sich zu speichern. Bei einem bestimmten Knopfdruck soll der Roboter das Loch für die jeweils nächste Form so ausrichten, damit der Mensch das Klötzchen nur noch einwerfen braucht.

3.2. Speichern der Anwendungsdaten

Um auf bestimmte Menschen zugeschnittene Bewegungsabläufe zu machen, muss der Roboter Daten über den Anwender kennen. Diese sollten persistent gespeichert werden, damit bei einem Wechsel des Anwenders die Daten nicht verloren gehen. Daten der Anwender sind z.B. Name, Alter, bestimmte Positionen im Roboterprogramm, etc.

3.2.1. Speichern über Polyscope und URScript

In der Polyscope Software oder in einem URScript Programm, können Daten, die von den Benutzern erstellt oder erhoben werden, nicht persistent gespeichert werden.

Hierzu muss eine Zweite Anwendung Entwickelt werden, auf die sich das URScript oder URP Programm verbindet und die Daten zum persistenten speichern versendet.

In Polyscope und URScript muss sehr aufwendig mit den vorhandenen Script Befehlen eine Socket Verbindung aufgebaut werden. Damit diese Zwei Programme miteinander Kommunizieren können muss ein gemeinsames Protokoll mit bestimmten Befehlen festgelegt werden.

3.2.2. Speichern über Eigene API

Mit der Eigenen API muss keine zweite Software entwickelt werden, da die API auf einem Client Rechner läuft und dort die Daten persistent gespeichert werden können. Es muss im Anwendungsprogramm eine verbindung zu einer Datenbank aufgebaut werden und dort können die Daten gespeichert werden.

4. Realisierung

4.1. TCP Server mit Datenbank zum dauerhaften speichern der Daten

Um mit Polyscope und URScript erhobene Daten zu Speichern wurde ein kleiner TCP Server geschrieben, der eine Verbindung zulässt und Daten in einer Datenbank Speichert. Die Daten sind Objectorientiert, und werden von dem Server erstellt. In Polyscope und URScript gibt es keine Objectorientierung, deshalb muss dort alles nacheinander angefragt werden.

4.2. C-API

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.2.1. gelöste Aufgabe

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et

dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.2.2. Aufgetretene Probleme

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.3. Polyscope

4.3.1. Programmierung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.3.2. Benutzer Interaktion

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et

dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.3.3. Test und Fehlersuche im Programm

Bevor Polyscope ein Programm ablaufen lässt wird das Script auf die richtige Syntax geprüft. Sollte ein Fehler vorhanden sein wird dies beim Start als Popup angezeigt. Fehler die in Abschnitten mit Touch hinzugefügt wurden, können jedoch nicht lokalisiert werden. Nur in extra eingefügtem Script code kann grob lokalisiert werden, was für ein Fehler aufgetreten ist, weil dieser Teil extra geprüft wird.

4.3.4. Aufwand der Programmierung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.4. URScript

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.4.1. Laden des Scripts auf den Controller

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.4.2. Programmierung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.4.3. Test und Fehlersuche im Programm

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.4.4. Benutzer Interaktion

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.4.5. Aufwand der Programmierung

4.5. Anwendung mit Eigener API

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.5.1. Adapter zur Secondary Schnittstelle

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.5.2. Programmierung mit Adapter

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.5.3. Benutzer Interaktion

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.5.4. Test und Fehlersuche im Programm

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.5.5. Aufwand der Programmierung

5. Ergebnis

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.1. Vergleich der Schnittstellen

TODO !! Tabellen vergleich

5.2. Nicht erreichte Ziele

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.2.1. C-API

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6. Fazit

6.1. Ausblick

[5.2](#) Nicht erreichte Ziele.

6.2. Zusammenfassung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6.2.1. C-API

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6.2.2. UR-Script

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6.2.3. Polyscope

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6.2.4. Eigene API

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6.2.5. Grobe Empfehlung, welche Schnittstelle, bei welchem Anwendungsfall verwendet werden soll

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Literaturverzeichnis

[DJAN-2013] Name1, Vorname1[; Name2, Vorname2; ...]: Titel [: Untertitel].
[Auflageneigenschaften] Verlags(kurz)bezeichnung, Verlagsort,
Jahr der Auflage. [S. xx - yy.] (Buch)

B. Glossar

BSD Berkeley Software Distribution: Software dieser Lizenz wird frei vermarktet und darf verändert, verbreitet und kopiert werden.

C. Quellcode

C.1. Speichern der daten über TCP in der Datenbank