Universität Kassel
Fachbereich 16 - Informatik und Elektrotechnik

Teamarbeit

Abschlussbericht

Autoren: Dennis Knitterscheidt
Robert Meschkat 28227496
Philipp Schenk 33309370
Eric Wagner 32233447

Betreuer: M. Sc. Stephan Opfer

Inhaltsverzeichnis

1 Einleitung

Philipp
REMOVE THIS: Beispiel

Kurze Beschreibung des Themas Teamarbeit. Was ist der Bericht?

2 Technische Arbeit

2.1 Arbeitsauftrag

Im KickOff-Workshop zur Veranstaltung „Teamarbeit“ wurden mehrere Aufgaben vorgestellt. Das Team hat sich dann zusammen gefunden, um folgende Aufgabe zu bearbeiten: Für die vom Team bearbeitete Aufgabe sollten ein Turtlebot und die existierende Software so erweitert werden, dass der Turtlebot als Transportsystem im Fachgebiet eingesetzt werden kann. Der Turtlebot sollte über die Software an einen Ort im Fachgebiet geschickt werden können, an dem er einen Gegenstand entgegen nimmt. Der Gegenstand sollte dann an einen vorher in der Software definierten Zielort gebracht werden.
Für die Umsetzung sollte ein Behältnis für den Transport der Gegenstände auf dem Turtlebot befestigt werden. Der Behälter sollte mit entsprechender Sensorik ausgestattet werden, damit der Turtlebot erkennt, wenn ein Gegenstand auf ihm plaziert wurde. Da der Turtlebot nur Daten verarbeiten kann, die sich in seinem Weltmodel befinden, mussten die Sensordaten außerdem in das bestehende Framework eingepflegt werden.
Um dem Turtlebot den Transportauftrag zu erteilen, sollte eine graphische Benutzeroberfläche entwickelt werden, in der man den Start- und Zielpunkt, sowie den zu transportierenden Gegenstand festlegen können soll.

2.2 Entwurft

2.2.1 Erste Bestandsaufnahme

Da kein Teammitglied zuvor mit den Turtlebots gearbeitet hat, musste sich das Team zunächst einen groben Überblick über die bestehende Hard- und Software verfassen, bevor mit der eigentlichen Planung begonnen werden konnte.
Für die Sensorik des Behälters war der Aufbau des Turtlebots ausschlaggebend: Gesteuert wird der Turtlebot von einem Notebook. Dieser ist über USB sowohl mit der Fahrbasis, als auch mit der Sensorik für die räumliche Erkennung verbunden. Damit wurde klar, dass auch der Gegenstandsdetektor über USB mit dem Notebook verbunden werden muss.
Die Software-Sammlung, die für den Betrieb des Turtlebots benötigt wird, basiert auf dem „Robot Operating System“, kurz ROS genannt, in der Version *Kinetic* und ist in C++ verfasst. Das User Interface musste als mit ROS kommunizieren können und im besten Fall in C++ geschrieben sein.

2.2.2 Recherche

2.2.3 Erster Entwurf

- Turtlebot beschreiben
  - Was ist der Turtlebot?

  - Was kann er?
  - Was wurde im Fachgebiet schon damit gemacht?
- Verwendete Soft- und Hardware
  - Verwendung von Linux 16.04 als Betriebssystem
  - Roboterprogrammierung in C++
  - Verschiedene existierende Repositories und GIT
  - Verwenden von QT oder Chromium für das Interface
  - Verwenden von Arduino mit Taster als Sensor
  - Rosserial Arduino zur Umwandlung in Nachrichten
  - Haribo-Korb zum Tragen
  - Steuerung des Roboters über Nachrichten aus der UI
  - Optionale Idee: RFID-Leser mit Tags in Tassen oder Büchern
- Vorwissen der Teammitglieder beschreiben (hier oder bei Teamrollen?)
  - Robert: Erfahrungen in Linux
  - Eric: Erfahrungen in UI-Programmierung
  - Dennis: Erfahrungen mit Tastern und Hardware

2.3 Umsetzung

Eric (Bis UI)

- Am Anfang Beschäftigung mit den Themen und Einarbeitung (Installation von Ubuntu und ROS)
- Besprechung mit dem Betreuer zu Konkretisierung des Auftrags
- Grafik vom 14.05. einbauen und beschreiben
- Entscheidung für QT und RQT für die UI (Beschreiben)
- Probleme durch Betriebssystem und Branches erwähnen
- Parallele Arbeit an Taster und UI
- Nachdem Laptop nicht funktioniert hat wurde auf den Rechner umgeschwenkt
- (Vielleicht UI und Taster in zwei Unterkapitel teilen)

2.3.1 UI

- Verwenden von QTCreator zum Erstellen des UI-Fensters
- Schreiben des Codes mit C++ und ROS (RQT)
- Erstellen der UI zu Hause
- Bugfixen auf dem Rechner als Team
- Nachrichtenart vom rviz Plugin entnommen (Pose_Stamped)
- Erste Version zeigen (Bild) und beschreiben
- Erste Version der Datenstruktur beschreiben
- Fehler in der UI-Entwicklung beschreiben und Verbesserungen sagen
- Error Handling bei schlechter Config Datei
- Umstellung auf existierende Kartendaten mit anderer Struktur (Karte entspricht nicht der Roboterkarte) (Robert)
- Verbesserung der UI mit den neuen Punkten (Zweite Version zeigen)

2.3.2 Taster

Dennis

- Besprechung verschiedener Taster
- Entscheidung für Arduino
- Schreiben des Codes und Bauen des Tasters
- Einbauen des Tasters in das Weltmodell (Philipp)
- Verbesserung des Tasters mit Korb aus Haribo

2.4 Ausblick

Eric

- Auftrag ist nicht ganz fertig geworden
- Schreiben eines Behaviours, das den Taster verwendet

- Verwenden von anderen Nachrichtentypen zum Senden
- Anpassen der Kartendaten mit Roboter-Infos
- Roboter kann per Text to Speech den gesuchten Gegenstand sagen
- Verwendung von RFID zur Erkennung der Gegenstände

3 Teamarbeit

3.1 Teamrollen

In der Einführungsveranstaltung wurden zwei Aufteilungen von Teamrollen vorgestellt: Teamrollen nach Basadur und nach Belbin.

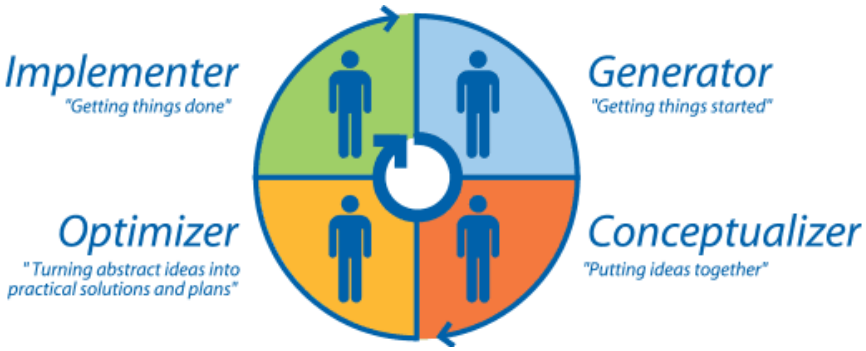

Abbildung 1: Teamrollen nach Basadur

Auf Abbildung 1 sind die Teamrollen nach Basadur zu sehen. Diese wurden in Generator, Conceptualizer, Optimizer und Implementer aufgeteilt. Der **Generator** ist derjenige, der Dinge ins Rollen bringt und viele Ideen zur Problemlösung sucht. Hierbei ist es schwer, ihn auf eine Idee festzunageln. Der **Conceptualizer** nimmt die Ideen vom Generator und versucht diese zusammen mit eigenen zu verwenden, um nach Lösungen zu suchen. Er versucht dabei, das Problem vollständig zu begreifen. Der **Optimizer** nutzt die abstrakten Ideen von Generator und Conceptualizer zur Umsetzung. Meist fokussiert er sich auf ein Problem und vertraut auf seine eigenen Fähigkeiten bei der Suche nach Lösungen. Der **Implementer** probiert Dinge lieber praktisch aus und verwirft Theorien, die nicht zu dieser Praxis passen. Er ist enthusiastisch und kann gut mit Menschen arbeiten. Dadurch wirkt er manchmal ungeduldig und aufdringlich.

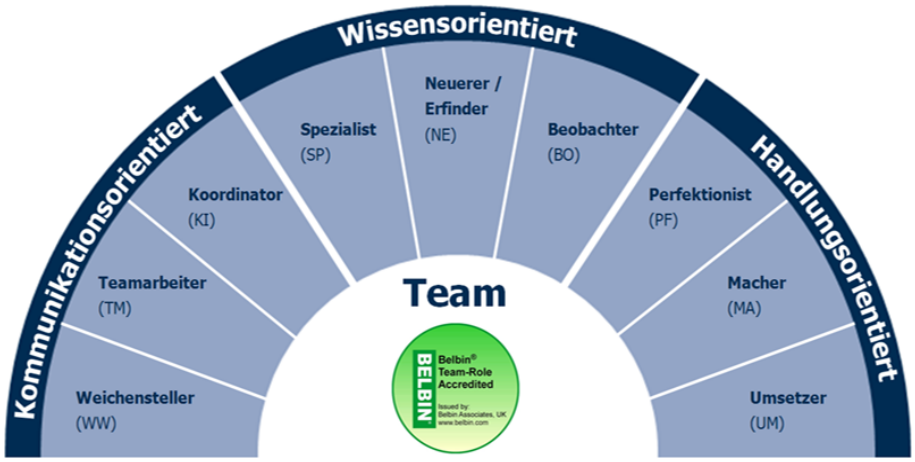

Abbildung 2: Persönlichkeiten im Team nach Belbin

Auf Abbildung 2 kann man die Teampersönlichkeiten nach Belbin sehen, diese sind in drei Kategorien unterteilt, sodass ein Teammitglied meist mehr als einen Typ hat. Im Folgenden werden die Rollen mit jeweils einem Satz beschrieben.
Der **Weichensteller** kommuniziert innerhalb und außerhalb des Teams.
Der **Teamarbeiter** ist anpassungsfähig und arbeitet flexibel mit den anderen.
Der **Koordinator** versucht, alle zum gemeinsamen Ziel zu bewegen.
Der **Spezialist** hat besondere Fähigkeiten, die er zum Lösen der Aufgabe gebrauchen kann.
Der **Erfinder** findet Ideen, die als Grundlage für die Arbeit verwendet werden können.
Der **Beobachter** überdenkt Aufgaben kritisch und sucht nach Lösungen.
Der **Perfektionist** achtet auf Details und verfolgt Dinge bis zum Ende.
Der **Macher** arbeitet hochmotiviert und möchte etwas erreichen.
Der **Umsetzer** hat einen Sinn fürs Praktische und geht systematisch vor.

In den folgenden Kapiteln haben sich die Teammitglieder selber und gegenseitig ein den Teamrollen nach Basadur und Belbin eingeschätzt.

3.1.1 Dennis

Basadur:
Belbin:

3.1.2 Robert

Basadur:
Belbin:

3.1.3 Philipp

Basadur:
Belbin:

3.1.4 Eric

In den Teamrollen nach Basadur schätzt sich Eric als **Conceptualizer** und **Optimizer** ein. Als Conceptualizer hat er zu Beginn des Projektes Ideen gesammelt und das Problem in Unterprobleme aufgeteilte. Er sammelte die Ideen und versuchte diese umzusetzen und zu überdenken. Beim Umsetzen der Ideen und Problemlösen hat er sich mit den anderen Teammitgliedern zusammengefunden und gemeinsam nach Lösungswegen und Lösungen zu suchen. Als Generator schätzt sich Eric weniger ein, da er sich meist auf eine Idee festlegt und selber selten zu Ideen kommt. Zusätzlich denkt er von sich nicht als Implementer, da er meist eher theoretisch über Ideen nachdenkt und Theorien aufstellt. Trotzdem kann er wie ein Implementer gut mit Menschen arbeiten.
Bei den Teamtypen nach Belbin sieht sich Eric als **Weichensteller** und **Koordinator**. Im Laufe der Teamarbeit hat er sich als Teamleiter etabliert, indem er die Arbeit bei den wöchentlichen Team-Treffen protokolliert hat. Dazu koordinierte er die Teammitglieder und versuchte bei Besprechungen jeden mit einzubeziehen. War ein Teammitglied nicht anwesend bei dem Treffen, so wurde diese Person von Eric entweder direkt danach durch die WhatsApp-Chatgruppe oder zum nachfolgenden Treffen informiert, was sie verpasst hatte. Zum Teil lässt sich Eric noch als **Spezialist** einschätzen, da er vor dem Projekt bereits mit ROS, Arduino und UIs gearbeitet hat.

3.2 Teamphasen

Dennis
Grafik der Phasen einbauen.

3.2.1 Forming

Bis wohin wurde nix geschafft? Teamfindung.

3.2.2 Storming

Aushilfe vom Betreuer. Sachen kompilieren und funktionieren.

3.2.3 Norming

Kickoff-Meeting.

3.2.4 Performing

Abschluss. Dokumentation.

3.3 Probleme und Lösungen

Philipp
Komplikationen mit dem Turtlebot

- Falsche Branches (Mit Betreuer gelöst)
- Akku kaputt (Tausch des Akkus durch Betreuer)
- Kommunikation nicht möglich (Deaktivieren eines Netzwerks)
- Kartendaten sind nicht akkurat auf den Turtlebot zugeschnitten

Am Protokol orientieren.

4 Fazit

(Gemeinsam, jeder ein Absatz?) Wie hat die Arbeit im Team funktioniert? Anwendung der Workshop-Sachen auf reale Teamarbeit.