

Übungsblatt 03: Architektur

Software Engineering (WiSe 24/25)

Abgabe: **Fr. 22.11.2024, 10:00 Uhr**

Besprechung: Fr. 15.11.2024

Fallstudie: Roboter-Fußball

Seit einigen Jahren wird viel Forschungsaufwand in die Entwicklung von Kontroll- und Entscheidungssystemen investiert, die mit ihrer Umgebung interagieren (vgl. cyber-physical systems). Populäre Beispiele sind u.a. autonome Roboter und autonomes Fahren. Um wiederkehrende Probleme zu identifizieren und passende Lösungen zu entwickeln, wurden konkrete Anwendungsdomänen vorgeschlagen. Eine dieser Domänen ist *Roboter-Fußball*, in dem zwei Teams autonomer Roboter in einer kleinen Version eines Fußballspiels gegeneinander antreten. Es existieren verschiedene Ligen, die u.a. anhand der teilnehmenden Art von Robotern unterschieden werden¹. Ein Beispiel ist die *RoboCup Small Size League*, die CUBE-Roboter einsetzen (siehe Abbildung 1).

Es ist Ihre Aufgabe, verschiedene Architekturstile zu betrachten und mit deren Hilfe eine Architektur für eine vereinfachte Variante dieser Anwendungsdomäne zu entwerfen und zu modellieren. Die Architektur soll einem Team von autonomen Robotern ermöglichen, gemeinsam Fußball zu spielen.

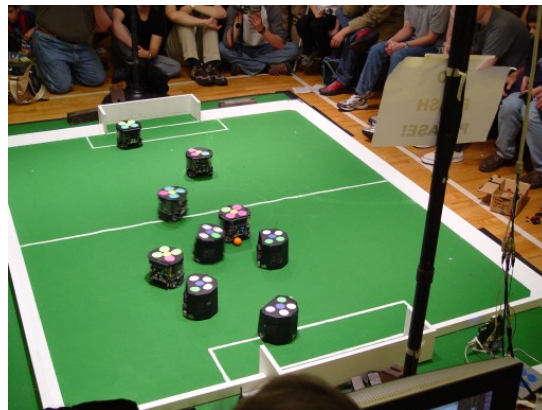
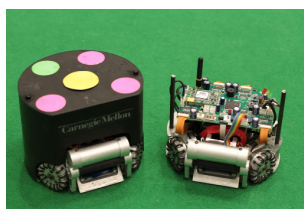


Abbildung 1: Beispiel eines Roboters und eines Roboter-Fußballspiels.

Das Ziel eines Teams ist es, mehr Tore als das gegnerische Team zu erzielen. Beiden Teams stehen dabei die gleichen *privaten Ressourcen* zur Verfügung:

- *Drei Roboter* (Anmerkung: bei RoboCup kommen häufig größere Teams zum Einsatz). Jeder der drei Roboter eines Teams ist eindeutig durch ein Muster farbiger Kreise an der Oberseite identifizierbar. Für die Kreise werden CMYK-Farben (cyan, magenta, yellow, black) verwendet, sodass sie nicht mit dem Spielfeld (grün) und dem Spielball (orange) verwechselt werden können. Die Roboter selbst übernehmen keine Berechnungen, sondern empfangen die für das Spiel notwendigen Informationen (z.B. neue Zielposition, die ein Roboter einnehmen soll) von der zentralen Team-Recheneinheit.

¹<https://www.robocup.org>

- *Eine zentrale Team-Recheneinheit*, die sowohl bidirektional mit jedem der Roboter einzeln kommunizieren, als auch Broadcast-Nachrichten an alle Roboter des eigenen Teams senden kann.

Beide Teams nutzen zusätzlich die folgenden *geteilten Ressourcen*:

- *Das Spielfeld*: alle Aktionen werden auf einer flachen Matte durchgeführt, die flüssige Roboterbewegungen zulässt. Die Matte ist gleichmäßig grün gefärbt, ausgenommen der weißen Markierungen. Insbesondere limitiert ein weißes $200\text{cm} \times 100\text{cm}$ großes Rechteck die Spielfläche. Die weiteren weißen Markierungen (z.B. der Mittelkreis und die Tormarkierungen) sind für das tatsächliche Spielgeschehen unerheblich.
- *Die Tore*: An beiden kurzen Seiten des Rechtecks ist jeweils der mittlere 40cm -Streifen mit einem unsichtbaren Sensor abgedeckt. Wann immer dieser Sensor den Ball erkennt, der die Linie zur Gänze überquert hat, wird dies an die zentrale Spiel-Recheneinheit gemeldet, die ein Tor für das entsprechende Team verbucht.
- *Der Spielball*: ein kleiner, leuchtend oranger Ball wird für das Spiel genutzt. Die Roboter bewegen sich entsprechend der Spielstrategie (vorgegeben durch die zentrale Team-Recheneinheit) zum Ball, um diesen dann möglichst zum gegnerischen Tor zu schießen.
- *Die zentrale Spiel-Recheneinheit*: Diese Recheneinheit verwaltet den Spielstand und erkennt Spielereignisse, wie z.B.: der Ball verlässt die Spielfläche, oder es wurde ein Tor erzielt. Alle diese Ereignisse werden den Team-Recheneinheiten *gleichzeitig* mitgeteilt.
- *Eine Vogelperspektiv-Kamera*: Diese Kamera nimmt kontinuierlich Bilder der gesamten Spielfläche in geringer Auflösung auf und leitet diese an die Team-Recheneinheiten und die zentrale Spiel-Recheneinheit weiter. Die Kamera stellt vier Bilder pro Sekunde zur Verfügung, d.h. ein Bild alle 250ms .

Es ist die Aufgabe der Team-Recheneinheit, alle Positionen der Spieler des Teams und die Position des Balls zu ermitteln, eine globale Spielstrategie zu entwickeln und die Roboter über den für sie wichtigen Teil dieser Strategie zu informieren. Um alle Spielerpositionen zu ermitteln, muss die Team-Recheneinheit ein Kamerabild entsprechend der CMYK-Farben mit vier Filtern (Komponenten) bearbeiten. Entsprechend wird auch ein weiterer (oranger) Filter für den Ball benötigt. Der Team-Recheneinheit stehen solche Filter zur Verfügung. Allerdings sind diese Filter kleiner als das eigentliche Bild: sie können jeweils nur $1/3$ des vollständigen Bilds gleichzeitig filtern. Weiterhin benötigt jeder Filter 500ms , um alle Berechnungen abzuschließen.

Sobald die Position der Spieler und des Balls erkannt wurden, muss eine Spielstrategie durch die Team-Recheneinheit entwickelt werden, um dann die entsprechenden Informationen (z.B. neue Zielposition jedes Roboters) an die jeweiligen Roboter zu übermitteln.

Alle diese Prozesse (Identifikation von Spieler- und Ball-Position, Strategieentwicklung und Kommunikation der Strategie an die Roboter) müssen kontinuierlich durchgeführt und so schnell wie möglich abgeschlossen werden.

Neben der Verarbeitung des Stroms an Bildern, reagiert die zentrale Team-Recheneinheit auf die Ereignisse "Tor ist gefallen" und "Ball verlässt die Spielfläche", über die sie von der zentralen Spiel-Recheneinheit informiert wird. Für diese Ereignisse werden vordefinierte Strategien verwendet, z.B. die Roboter nehmen nach einem Tor die übliche Startaufstellung ein.

Die zentrale Spiel-Recheneinheit identifiziert das Ereignis "Tor ist gefallen" über eine Nachricht eines Sensors, der in jedem Tor verbaut ist und das Übertreten der Torlinie durch den Ball erkennt. Das Ereignis "Ball verlässt die Spielfläche" wird durch die Analyse der Bilder der Vogelperspektiv-Kamera identifiziert, wozu ein oranger Filter notwendig ist, um die Position des Balls festzustellen – identisch zu dem Filter, den die Team-Recheneinheiten verwenden.

Alle Recheneinheiten (Spiel- und Team-Recheneinheiten) teilen sich keine Ressourcen und Komponenten, d.h. jede Recheneinheit hat z.B. ihre eigenen Filter. Die Recheneinheiten unterliegen keiner Beschränkung der Hardwareressourcen (z.B. CPU und RAM), um parallele Verarbeitungsschritte zu ermöglichen. Hierzu können Komponenten desselben Typs mehrfach auf einer Recheneinheit instantiiert werden.

Aufgabe 1: Identifikation der Komponenten und deren Funktionalität (39 Punkte)

Ihre Aufgabe ist es, eine Architektur für diese Anwendungsdomäne zu entwickeln, wobei Sie lediglich **ein** Team berücksichtigen müssen. Daher besteht das System aus folgenden Subsystemen, die Softwarekomponenten haben:

- Drei Roboter
- Eine zentrale Team-Recheneinheit
- Zwei Sensoren, einer für jedes Tor des Spielfelds
- Eine zentrale Spiel-Recheneinheit
- Eine Vogelperspektiv-Kamera

Für den Entwurf der Architektur betrachten Sie **die drei folgenden Szenarien** jeweils vollständig (d.h. von Start bis Ende):

1. “Tor!”
Start: Ball übertritt die Torlinie.
Ende: Die Roboter des Teams nehmen ihre vordefinierte Startpositionen ein.
2. “Ball im Aus”
Start: Kontinuierliche Aufnahme von Bildern durch die Vogelperspektiv-Kamera und der Ball übertritt die Spielfeldmarkierung.
Ende: Die Roboter des Teams nehmen ihre vordefinierte Positionen um den Einwurfpunkt ein.
3. “Spielstrategie”:
Start: Kontinuierliche Aufnahme von Bildern durch die Vogelperspektiv-Kamera.
Ende: Roboter empfangen kontinuierlich neue Zielpositionen und steuern diese an.

Für jedes der drei Szenarien:

- (a) (15 Punkte) **Beschreibung des Szenarios:** Beschreiben Sie in einer Liste von maximal 6 Teilschritten wie das Szenario zu realisieren ist. Sie können Stichpunkte verwenden. (5 Punkte je Szenario)

Hinweise:

- Für die gesamte Aufgabe könnte sich auch eine Tabelle pro Szenario anbieten.
- Eine Verfeinerung (Reduktion der Abstraktion) von Teilschritten sollte *nicht* erfolgen, um die Komplexität der Szenarien in Grenzen zu halten. Zum Beispiel kann die Entwicklung einer Spielstrategie basierend auf den Positionen der Spieler und des Balls als *ein* Teilschritt betrachtet werden.

- (b) (12 Punkte) **Subsysteme und Komponenten:** Weisen Sie jedem Teilschritt aus (a) ein Subsystem zu (Roboter, Zentrale Team-Recheneinheit, Tor, Zentrale Spiel-Recheneinheit, Vogelperspektiv-Kamera), das diesen Teilschritt realisiert. Identifizieren Sie hierbei mögliche Komponenten, die die Softwareanteile dieses Teilschritts realisieren. (4 Punkte je Szenario)

Hinweise:

- Im Allgemeinen kann ein Teilschritt durch ein oder mehrere Komponenten realisiert werden.

- (c) (6 Punkte) **Abhängigkeiten zwischen Subsystemen:** Diese Aufgabe ist in der Moodle Übung 03 zu beantworten. (2 Punkte je Szenario)

- (d) (6 Punkte) **Nicht-funktionale Anforderung:** Diese Aufgabe ist in der Moodle Übung 03 zu beantworten. (2 Punkte je Szenario)

Aufgabe 2: Entwurf und Modellierung der Architektur (61 Punkte)

Entwerfen und modellieren Sie eine Architektur mit einem UML-Komponentendiagramm, das Subsysteme, Komponenten, Schnittstellen und Konnektoren umfasst (Geben Sie allen Elementen des Diagramms Namen). Bitte erstellt ein UML-Komponentendiagramm pro Szenario. Überprüfen Sie, ob Architekturstile (Layers, Pipes & Filters, Peer-to-Peer, und Blackboard) auf Teilprobleme der Szenarien anwendbar sind und verwenden Sie ggf. diese Stile beim Entwurf und der Modellierung. Begründen Sie kurz Ihre Wahl bzw. Nichtwahl eines Stils.

Hinweise:

- Beachten Sie die Hinweise zur Architekturmodellierung aus der Übung.