

# Praktikum Software Engineering 1

---

## 4. Analyse und Design

### Learning Outcome des Praktikums

Im Rahmen dieses Praktikumstermins soll die Analyse- und Design-Kompetenzen trainiert werden. Hierzu ist eine Systembeschreibung mittels der in der Vorlesung vorgestellten Heuristiken zu analysieren und ein erstes Software Design für die Steuerungssoftware zu erstellen. Das Design kann nicht vollständig erstellt werden, da die Heuristik nur Hinweise gibt und die Lücken im Design in einem weiteren Schritt geschlossen und in ein vollständiges Design überführt werden müssen. Neben der Modellierung ist auch die Auswahl einer geeigneten Modellierung Ziel dieses Praktikums.

### Ablauf des Praktikums

Führen Sie **vor dem Praktikumstermin** in ihrem Team eine Analyse des gegebenen Falls mit Hilfe der Heuristiken durch. Auf Basis der Analyse sollen identifizierte Designelemente in den entsprechenden UML-Diagrammen entworfen werden. Zu jedem Element (Klasse, Interface usw.) soll jeweils kurz die Aufgabe/Verantwortlichkeit dokumentiert werden. Zum Beispiel:

Klasse Kundendatenbank: Speichert und verwaltet die Kundendaten. Erstellt Reports.

Behörden-Schnittstelle: Bereitstellung aller Kundendaten nach XML-BhKD-Protokoll auf Anfrage.

**Im Praktikum** erfolgt dann eine Überprüfung der Analysen und des Designs durch eine vereinfachte Inspektion im Plenum.

### Deliverables des Praktikums

- Modelle
- Textdokument mit ergänzenden Erläuterungen

### Fallbeispiel mobiler Roboter

Ein mobiles Robotersystem mit Roboterarm wird auf der Basis einer zweirädrigen Plattform aufgebaut. Die Plattform ist ähnlich dem Pioneer 3 DX.

Aufgabe des Robotersystems ist es, autonom Arbeitstische anzufahren, die als Zielkoordinaten vorgegeben sind<sub>11</sub>. Dort wird die auf der Plattform montierte Kamera auf die Arbeitsfläche ausgerichtet<sub>12</sub>. Eine Objekterkennung wird durchgeführt<sub>13</sub>. Jedes Objekt auf der Arbeitsfläche wird durch den auf der Plattform montierten Roboterarm gegriffen und näher an die Kamera geführt<sub>14</sub>. Die Kamera erstellt von dem Objekt Fotos aus mehreren Perspektiven<sub>15</sub>. Die Fotos werden gespeichert und später offline zur 3D-Rekonstruktion verwendet<sub>16</sub>. Zusammen mit den Fotos wird das Gewicht des gehobenen Gegenstandes abgespeichert<sub>17</sub>.

Die Basisplattform verfügt über zwei einzeln angetriebene Räder<sub>21</sub>. Jedes Rad wird über einen Elektromotor mit Leistungselektronik direkt angetrieben<sub>22</sub>. Die Drehrichtung und der zurückgelegte

Drehwinkel eines jeden Rades wird über Inkrementalgeber, mit 500 Pulsen pro Umdrehung, erfasst<sup>23</sup>. Ein Impulszähler übernimmt die Erfassung der Pulse in der Embedded Control Unit (Steuerungscomputer, Mikrocontroller)<sup>24</sup>. In der ECU gibt es auch einen Peripherie-Baustein für die Ansteuerung der Leistungselektronik für jedes Rad<sup>25</sup>. Über den Peripherie-Baustein wird der Strom für die Elektromotoren eingestellt<sup>26</sup>. Die Regelung in der Software muss die Inkrementalgeber-Signale auswerten und die Geschwindigkeit und Strecke entsprechend regeln<sup>27</sup>.

Aus den gezählten Pulsen kann der zurückgelegte Weg des Roboters ermittelt und seine Position geschätzt werden<sup>31</sup>. Die Positionsdaten werden an die Navigation weitergegeben<sup>32</sup>. An die Navigation werden auch die Daten der Entfernungsmesser, die im Winkel von 30° rund um den Roboter angebracht sind, weitergegeben<sup>33</sup>. Die Entfernungsmesser liefern den Abstand zu einem Gegenstand im Bereich von 20-110 cm<sup>34</sup>. Ist die Entfernung größer als 110 cm, so wird dieser Maximalwert geliefert<sup>35</sup>. Die Entfernungsmesser sind über CAN angeschlossen und senden die aktuell gemessene Entfernung alle 100 ms<sup>36</sup>.

Die Navigation erstellt aus den Positionsdaten und der Entfernungsmessungen eine Karte (SLAM)<sup>41</sup>. Eine initiale Karte kann geladen und später als aktualisierte Karte wieder gespeichert werden<sup>42</sup>. Als Teil der Navigation kann eine Route zu Zielkoordinaten berechnet werden<sup>43</sup>. Die Wegpunkte werden als Liste mit anzufahrenden Koordinaten und einzunehmenden Winkel im Wegpunkt geliefert<sup>44</sup>. Die Robotersteuerung muss die Motoren so ansteuern, dass der jeweils nächste Wegpunkt angefahren wird<sup>45</sup>.

Der Roboter darf sich nach dem Einschalten erst dann bewegen, wenn die Initialisierung der Leistungselektronik und die Selbstkalibrierung der Sensoren abgeschlossen ist<sup>51</sup>. Wird die Leistungselektronik zu früh angesprochen, so kann sie beschädigt werden<sup>52</sup>. Die Sensoren liefern während der Selbstkalibrierungsphase die Kalibrierungswerte, die nicht als Entfernungswerte angesehen werden dürfen<sup>53</sup>. Stellt die Leistungselektronik eines Roboters einen Fehler fest, dann muss durch die Software auch der andere Elektromotor abgeschaltet werden<sup>54</sup>.

Der auf der Plattform montierte Roboterarm ist ein 3-Achs-Knickarm<sup>61</sup>. Die Achsen verfügen über eine voll integrierte Elektromechanik mit Harmonic-Drive Getrieben<sup>62</sup>. Über CAN kann der einzunehmende Winkel und die maximale Winkelgeschwindigkeit bei Positionsänderungen übertragen werden<sup>63</sup>. Die Achsen werden autonom ausgerichtet<sup>64</sup>.

Die Roboterhand ist ein einfaches Modell, welches zugreifen, sich drehen und horizontal (Handgelenk) abknicken kann<sup>71</sup>. Die Roboterhand ist ebenfalls mit der Leistungselektronik voll integriert und kann über CAN angesteuert werden<sup>72</sup>. Über CAN wird von der Hand das aktuell gemessene Gewicht des Gegenstandes gemeldet<sup>73</sup>.

Der aktuelle Zustand des gesamten Robotersystems wird über ein Display an der Plattform angezeigt<sup>81</sup>. Dort kann der aktuelle Batteriezustand abgelesen werden<sup>82</sup>. Eine Visualisierung der Entfernungsmessungen (jeder Sensor) wird grafisch dargestellt<sup>83</sup>. Alle aktuellen Sensorwerte, Betriebszustände und die aktuell erstellte Karte sollen über eine Web-Schnittstelle abrufbar sein<sup>84</sup>.

## Aufgaben

Modellieren Sie auf Basis der Fallbeschreibung das System. Wählen Sie für die in der Fallanalyse gefundenen Aspekte geeignete Darstellungen und Ebenen.

### Systemstruktur ermitteln

Gehen Sie den Text durch und bestimmen Sie die Hardware-Komponenten aus denen der Roboter und ggf. die ECU besteht. Ermitteln Sie, welche Komponenten miteinander interagieren. Erstellen Sie zur Übersicht ein Blockdiagramm.

### Detaillierte Textanalyse

Ihrem Team wird ein Textabschnitt (Nummernbereich mit Subscript-Zahlen an den Sätzen) zur detaillierten Analyse zugewiesen. Für jeden der Sätze leiten Sie entweder

- Ein formales Modell oder ein Fragment eines Modells ab. Das Modell kann eine mathematische Formel, Bool'sche Formel, ein Teil eines UML-Diagramms oder ähnliches sein. Pseudo-Code ist nicht erlaubt.
- Oder eine Konsequenz, auf die sie bei der Modellierung oder Implementierung achten müssen. Die Konsequenz ist textuell zu dokumentieren.

Versuchen Sie so exakt wie möglich zu sein!

### Datenmodell ermitteln

Gehen Sie den gesamten Text durch und bestimmen Sie, welche Daten in dem Software-System verarbeitet werden und welche Zusammenhänge es hier gibt.

### Software Design erstellen

Erstellen Sie für das gesamte Softwaresystem ein erstes Software Design unter Verwendung der Stereotypen Boundary, Control, Service, View usw. Überlegen Sie auch, ob Sie Design Pattern verwenden wollen. Dokumentieren Sie die Verantwortlichkeiten für die einzelnen Klassen. Sollten Sie Abläufe identifizieren, erstellen Sie dazu auch die Verhaltensbeschreibungen.

## Bearbeitungshinweis

### Modellierung

Beim Modellieren können Sie sich in Details verlieren, was den zeitlichen Umfang des Praktikums sprengt. Konzentrieren Sie sich zunächst auf die Analyse (Textmarker/Buntstifte?). Leiten Sie daraus die direkt erkennbaren Elemente ab. Überlegen Sie jeweils, ob es sich um funktionale, logische, Ablauf-, Daten- oder Hardware-Aspekte handelt. Ergänzen Sie dann sinnvoll weitere notwendige Elemente und Interfaces in Ihrem Design. Identifizieren Sie, welche Art von Diagrammen Sie benötigen. Starten Sie ggf. mit einem Systemkontextdiagramm.

Beachten Sie dann, welche Elemente der Hardware sich in Software-Komponenten widerspiegeln.

## Arbeitsverteilung

Arbeiten Sie Time-Boxed, d.h. setzen Sie sich eine maximale Zeit für die Vorbereitung. Sie können innerhalb der Teams nach folgenden Aspekten parallelisieren:

- Jedem Teammitglied werden Abschnitte des Textes zugeordnet. Alle Design- und Diagrammfragmente werden extrahiert und später zusammengesetzt.
- Jedem Teammitglied werden einzelne Analyse/Designziele zugeordnet.

Prüfen Sie anschließend auf Konsistenz der Modelle!

## Qualitätskontrolle

Jede Aussage/Erläuterung im Fallbeispiel sollte sich im Design oder in der zugehörigen Dokumentation wiederfinden lassen. Eine Fragestellung zur Prüfung könnte sein: „Wie haben Sie den Satz x des Fallbeispiels in ihrem Modell abgebildet?“

## UML-Tool

Die Erstellung von Diagrammen kann auch auf Papier erfolgen. Für das Review im Praktikum müssen die Diagramme in einer elektronischen, beamerfähigen Form vorliegen.