

# Praktikumsbeschreibung

## Überblick:

Patientenmonitore sind technisch hoch-komplexe Systeme. Deren (z.T. sicherheits-kritischen) Funktionen hatten wir bereits im funktionalen Viewpoint kennengelernt und diese auch modelliert. Eines dieser Funktionen ist das *Measurement*, welches bei durch entsprechende Hardware (z.B. Sensoren) und SW (z.B. Messsoftware) die unterschiedlichen Werte des Patienten darstellt.

Um das System technisch zu „engineeren“, lernen wir in diesem Praktikum eine *technische Architektur* des Systems zu erstellen, sowie die Übergänge aus dem funktionalen und logischen Viewpoint zu modellieren.



## Technische Realisierung des Patientenmonitorsystems der Firma Philips

## Praktikum 5

In diesem Versuch modellieren Sie im Werkzeug *Cameo Systems Modeler* eine technische Architektur und die Verbindung folgender Viewpoints: Funktional zu Logisch und Logisch zu Technisch.

### Ziel:

Die Erstellung eines durchgängigen Architekturmodell (als Strukturmodell) des Patientenmonitors bestehend aus verbundener funktionaler, logischer und technischer Architektur. Dazu wird die Funktionalität Measurement (aus dem Functional Viewpoint) zuerst als logische Komponenten dargestellt (noch Plattform-unabhängig) und schließlich auch in der technischen Architektur mittels einer Software- und Hardware-Architektur modelliert. Die Modelle der verschiedenen Viewpoints werden im Anschluss über SysML Allokations-Matrizen miteinander verbunden.

### Aufgabe 1: Organisation der Technischen Architektur

Erweitern Sie präferiert Ihr bisheriges Modell des Patientenmonitors oder laden Sie sich die Vorlage aus dem ILIAS-Ordner herunter und öffnen Sie diese. Führen Sie folgende Schritte durch:

- Erstellen Sie einen Ordner (SysML Package) für die Technische Architektur, welchen Sie „40\_Technical“ nennen.

- Erstellen Sie in diesem Ordner ein BDD. In diesem BDD erzeugen Sie ein Package, das Sie „*Technical Architecture*“ nennen. Diesem Package fügen Sie zwei weitere Packages „*SW Architecture*“ und „*HW Architecture*“ hinzu.
- In der Containment-Ansicht (links im MSOSA) finden Sie die neuen Packages. Fügen Sie sowohl dem *SW Architecture*-Package, als auch dem *HW Architecture*-Package jeweils ein BDD hinzu.
- Beide BDDs sollen jeweils einen SysML Block, also einen übergeordneten Block, enthalten, der entsprechend die Namen „*SW Architecture*“ und „*HW Architecture*“ trägt.

## Aufgabe 2: Strukturmodellierung der Logischen Architektur

Seien Sie der Systems Engineer! Im Folgenden erhalten Sie einige Freiheiten diese Aufgabe zu bearbeiten, damit Sie das System nach Ihren Vorstellungen „engineeren“ können. Konkret heißt das, dass Sie entscheiden müssen, welche Komponenten der logischen Architektur anlegt werden sollen. Sie definieren also, welche logischen Komponenten zwei der 3 oben genannten Measurement – Funktionen (aus der funktionalen Architektur): „Measure Temperature“, „Measure Blood Pressure“ und „Measure SpO2“ später realisieren und somit in der logischen Architektur zu modellieren sind. (ggf. enthält Ihr Modell dieses schon).

Dabei ist das Folgende besonders wichtig: Sie müssen überlegen, welche der erstellten logischen Komponenten später in **Hardware** (z.B. Sensoren, usw.) realisiert werden und damit auch Artefakte im BDD *HW-Architecture* der technischen Architektur zugewiesen bekommen, und welche der neu erstellten logischen Komponenten zur Realisierung der jeweiligen Funktionen als **Software** realisiert werden und Artefakte im BDD *SW-Architecture* der technischen Architektur (als bspw. Tasks) zugewiesen bekommen.

Beispielsweise könnte die „*Measure-Temperature*“-Funktion durch 2 logische Komponenten realisiert sein, die in der technischen Architektur dann einen Hardware-Anteil (bspw. einen *temperatureSensor*) und einen Software-Anteil (bspw. *temperatueCalculation*) hat.

Schauen Sie in den Ordner für die Logische Architektur und öffnen Sie das BDD. Erstellen Sie nun die logischen Komponenten. Führen Sie daher folgende Schritte aus:

- Dekomponieren Sie die logische Komponente **Patient Monitoring System**, in so viele Subsysteme, dass ausreichend logische Komponenten zur Realisierung der 2 von Ihnen ausgewählten Funktionen vorhanden sind. Dabei soll **jede der 2 Funktionalitäten durch min. 2 logische Komponenten** realisiert werden:
  - Min. eine logische Komponente, die später (im BDD *HW-Architecture* der technischen Architektur) in Hardware realisiert wird
  - Min. eine logische Komponente, die später (im BDD *HW-Architecture* der technischen Architektur) in Software realisiert wird.
- Führen Sie auch eine Dekomposition der Ports durch.

## Aufgabe 3: Strukturmodellierung der Technischen Architektur

Öffnen Sie das oben erstellte BDD mit „*SW-Architecture*“ und „*HW-Architecture*“. Führen Sie folgende Schritte durch:

- Legen Sie eine SW-Architektur entsprechend Ihrer log. Komponenten aus Aufgabe 2 an: Das heißt konkret, dass Sie für die logischen Komponenten, die in Software realisiert werden sollen, im BDD „*SW-Architecture*“ entsprechende Tasks (als SysML Blöcke) mit Schnittstellen (als Proxy-Ports) anlegen. Ein IBD ist optional (aber nicht zwingend notwendig).
- Verbinden Sie diese Blöcke mit einer „Direct Composition“ mit dem übergeordneten Block „*SW Architecture*“, den Sie – wie oben beschrieben – angelegt haben.
- Führen Sie gleiches entsprechend für die „*HW-Architecture*“ durch.
- Verbinden Sie die angelegten Blöcke mit einer „Direct Composition“ mit dem übergeordneten Block „*HW Architecture*“, den sie – wie oben beschrieben – bereits angelegt haben.

- Legen Sie in der „*HW Architecture*“ zusätzlich einen SysML-Block „CAN-Bus“ an, den Sie auch mit einer „Direct Composition“ mit dem übergeordneten „*HW Architecture*“-Block verbinden.
- Erstellen Sie für die „*HW-Architecture*“ ein **IBD**, dass dann der **HW-Topologie** Ihres Systems entspricht.

#### Aufgabe 4: Verbinden von Viewpoints mittels Allokationen

Führen Sie folgende Schritte durch:

- Erstellen Sie zwei Allokationsmatrizen (SysML Allocation Matrix):
  - Die erste erstellen Sie im Package: *30\_Logical*
  - Die zweite erstellen Sie im Package: *40\_Technical*
- Die erste Allocation-Matrix soll den Übergang von Modellen des Funktionalen Viewpoints zu Modellen des Logischen Viewpoint darstellen:
  - Mapping von funktionalen Blöcken zu logischen Blöcken
  - Mapping von funktionalen Ports zu logischen Ports  
(vgl. hierzu Vorlesungsfolie MbSE\_09\_Technical\_Viewpoint Nr. 19 ff.)
- Die zweite Allocation-Matrix soll den Übergang von Modellen des Logischen Viewpoints zu Modellen des Technischen Viewpoint (konkret zu der entsprechenden SW-Architecture und HW-Architecture) darstellen:
  - Mapping von logischen Blöcken zu (entweder SW-Architecture oder HW-Architecture) Blöcken
  - Mapping von logischen Ports zu Ihren entsprechenden technischen Ports

#### Aufgabe 5:

Laden Sie Ihr Modell in den ILIAS-Abgabeordner hoch.

---