

Übungen zu „Softwarearchitekturen und modellgetriebene Softwareentwicklung“

Aufgabe 1 Klassifikation von Anforderungen [LÖSUNG]

Anforderungen kann man gemäß verschiedener Anforderungstypen strukturieren. Je nachdem welchem Anforderungstyp eine Anforderung zugeordnet wird, wird sie im nachfolgenden Entwicklungsprozess spezifisch modelliert.

Identifizieren Sie aus der Spezifikation des Drehzahlmessers (engl.: rev meter) aus dem Projekt EMPRESS (siehe Dokument „Empress: Auszug aus der Spezifikation eines Instrument Clusters“) beispielhaft einzelne Anforderungen und ordnen Sie diese den Anforderungsklassen der Business Requirements, Functional Requirements, System Requirements und HW Requirements zu.

Beispiele für die einzelnen Anforderungsklassen:

- Business Requirements:
Comfortable view of the rev meter, which provides actual information permanently
- Functional Requirements:
The error display lights up whenever an error is realized.
- System Requirements
Deployment of Model Y components: Rev Meter Model X has to be developed using existing components of Model Y.
- HW Requirements:
Use CAN-Bus

Aufgabe 2 Sichtenbasierte Modellierung von Anforderungen [LÖSUNG]

Bei der modell-basierten Anforderungsanalyse werden Anforderungen formal modelliert. Dabei kann man bei der Modellierung verschiedene Sichten auf eine Problemstellung und ein System unterscheiden (Interaktionssicht, Datensicht, Verhaltenssicht und Struktur-sicht), die jeweils einen anderen Aspekt des schrittweise zu spezifizierenden Systemverhaltens darstellen. Diese Sichten sollten am Ende der Anforderungsanalyse konsistent sein, d.h. sie dürfen keine widersprüchlichen Informationen enthalten. Mit Hilfe der Modelle ist es zusätzlich möglich, fehlende Anforderungen zu identifizieren und heraus zu arbeiten.

Erstellen Sie am Beispiel des Drehzahlmessers aus dem Projekt EMPRESS (siehe Dokument „Empress: Auszug aus der Spezifikation eines Instrument Clusters“) verschiedene

Anforderungsmodelle, die die oben genannten Sichten darstellen. Gehen Sie dabei vom UseCase „Show RPM of the Engine“ aus. Identifizieren Sie Konsistenzbedingungen zwischen den Modellen der verschiedenen Sichten und identifizieren Sie außerdem fehlende und widersprüchliche Anforderungen.

Sie können dazu das Werkzeug „AutoFOCUS2“ (<http://www4.in.tum.de/~af2/>) verwenden.

Lösungsvorschlag:

- Interaktionssicht: Siehe Abbildung 1

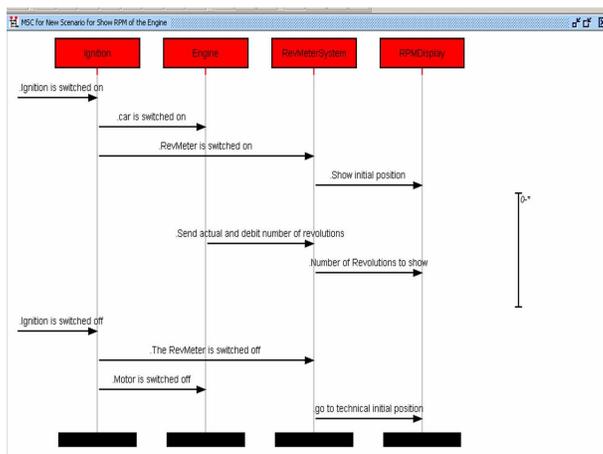


Abbildung 1: Sequenzdiagramm zu USE-Case „Show RPM of the Engine“

- Datensicht: Aus dem Lastenheft können folgende Daten (siehe 1.2.4.1) und Ihre Wertebereiche gewonnen werden.

Ausgetauschte Nachrichten:

– Inputs:

Lg_lock = active, inactive / 1,0 (Einmal als active und inactive definiert in 1.2.4.2.3; in 1.2.4.2.6 als 1 und 0 festgelegt)

Actual_Number_Of_Revolutions = 0, 1, ..., 255 (siehe 1.2.4.2.6)

Debit_Number_Of_Revolutions = 0,1, ..., 255 (siehe 1.2.4.2.6)

– Outputs:

Number_Of_Revolutions_Shown (wird im Lastenheft nicht weiter spezifiziert)

Warning_RM (wird im Lastenheft nicht weiter spezifiziert)

Error_RM (wird im Lastenheft nicht weiter spezifiziert)

– Lokale Variablen:

Damping_PT1 = 0,..., max (siehe 1.2.4.2.8, der Wert max ist in dem Lastenheft nicht beschrieben!)

Variant_Car = 0,1 (0=diesel, 1=benzin, siehe 1.2.4.2.8)

Release_Bit = 0,1 (siehe 1.2.4.2.8)

- Verhaltenssicht (High-Level): Siehe Abbildung 2
- Struktursicht (logische Struktur): Siehe Abbildung 3

