Implementierungsdokumentation

Definition und Durchführung von Messwertverarbeitung für den Physikunterricht auf Basis eines Raspberry Pis

Version 1.0.0

David Gawron Stefan Geretschläger Leon Huck
Jan Küblbeck Linus Ruhnke

10. August 2019

Inhaltsverzeichnis

1 Ziel der Implementierungsdokumentation

2 Ausarbeitungsstand der Abnahmekriterien

Im Folgenden werden die Muss-, Soll- und Wunschkriterien aus dem Pflichtenheft herangezogen, das in der ersten Phase des Projekts entstanden ist. Es findet eine Bestandsaufnahme statt, inwieweit die Kriterien erfüllt sind.

Falls das Softwareprodukt ein Musskriterium nicht wie im Pflichtenheft beschrieben aufweist, so führt dieses Dokument detailliert die Ursachen und Gründe hierfür auf. Falls das Softwareprodukt ein Sollkriterium nicht wie im Pflichtenheft beschrieben aufweist, so beschreibt dieses Dokument zwar nicht in jedem Detail, aber hinreichend informativ die Ursachen und Gründe hierfür. Nicht umgesetzte Wunschkriterien werden lediglich benannt, aber nicht hinterfragt.

2.1 Entwurf von Messkonfigurationen

Es fand ein Fallback statt. Die Messkonfigurationen werden nicht wie gewünscht graphisch durch ein Drag- and Drop Feld erstellt, sondern müssen textuell eingegeben werden. Dadurch verändert sich auch die Betrachtung, wie und ob die folgenden Kriterien überhaupt erfüllt werden können.

MK 1 Das Musskriterium „Hinzufügen eines Bausteins aus dem Prototypen-Feld zu der Messkonfiguration“ ist TODO

MK 2 Das Musskriterium „Anpassen von wichtigen funktionalen Bausteineigenschaften“ ist TODO

MK 3 Das Musskriterium „Löschen eines Bausteins aus der Messkonfiguration“ ist erfüllt, da der Benutzer die Textuelle Repräsentation eines Bausteins aus der Messkonfiguration entfernen kann.

MK 4 Das Musskriterium „Erstellen einer Verbindung“ ist umgesetzt. Der Benutzer kann ein Kanaltupel der Liste an Verbindungen hinzufügen und somit eine Verbindung der Messkonfiguration hinzu fügen.

MK 5 Das Musskriterium „Löschen einer Verbindung“ ist erfüllt. Der Benutzer kann eine Verbindung aus der Liste der Verbindungen löschen, in dem er das entsprechende Kanaltupel löscht.

SK 1 Das Sollkriterium „Undo-Redo-Funktion“ ist nicht umgesetzt. Die Messkonfiguration wird textuell erstellt und der Editor unterstützt keine Undo-Redo-Funktion.

WK 1 Das Wunschkriterium „Hinzufügen, Bearbeiten und Löschen von ergänzender Informationen zu der Messkonfiguration durch den Benutzer“ ist nicht umgesetzt.

2.2 Handhabung von Bausteinprototypen

SK 2 Der Benutzer ist in der Lage die Eigenschaften der Bausteinprototypen einzusehen. Jedoch ist die Ansicht auf das Anzeigen der *allgemeinen Bausteinprototyp-Informationen* beschränkt. Der Grund hierfür ist die Anbindung von dem Model an die GUI. Dadurch ist es aktuell nur möglich mit den allgemeinen Bausteinprototyp-Informationen zu arbeiten. Dementsprechend ist das Anzeigen der speziellen Eigenschaften, der Bausteinprototypen, nicht möglich.

SK 3 Das Kopieren der Bausteinprototypen ist möglich. Auch das Anpassen der allgemeinen Eigenschaften ist möglich. Jedoch können keine generischen Bausteinprototypen erstellt werden. Dafür wäre die Erstellung einer allgemeinen Vorläge nötig gewesen. Diese Erweiterung hätte dem Nutzer jedoch keine weitere Funktionalität geboten. Aus diesem Grund haben wir uns dazu entschieden die Funktion erst in einer eventuellen Erweiterung der Anwendung zu integrieren.

WK 2 Die Verwendung von erweiternder Software, über eine Schnittstelle, ist nicht mehr vorgesehen. Externe Software kann weiterhin zur Erstellung von Yaml-Dateien genutzt werden. Die so entstandenen Yaml-Dateien können über die Lade-Funktion der Anwendung aufgerufen und anschließend verwendet werden.

SK 4 Die Anwendung unterscheidet in ihrer jetzigen Form nicht zwischen Benutzerdefinierten- und System-Bausteinen. Deshalb ist es auch hier nur möglich die allgemeinen Eigenschaften der Bausteinprototypen zu ändern.

SK 5 Das Löschen von erstellten Bausteinprototypen ist ermöglicht.

2.3 Gewährleisten von Persistenz

MK 6 Die verwendeten Bausteine und ihre Anordnung, also die Messkonfiguration, kann in einer Datei gespeichert werden.

MK 7 Messkonfigurationen können aus Dateien geladen werden.

SK 6

SK 7

2.4 Bereitstellung vorgefertigter Teile

2.5 Handhabung von Messläufen

2.6 Benutzbarkeit der GUI

2.7 Abgrenzungskriterien

3 Umsetzung des Entwurfs

Während der Entwurfsphase wurden sowohl UML-Klassendiagramme als auch UML-Sequenzdiagramme erstellt. Zusammen mit der textuellen Beschreibungen der zu erstellenden Software-Elemente bildeten diese die Basis für die Produktion des Quellcodes während der Implementierungsphase.

In aller Regel lassen sich abstrakte Entwurfsinhalte während der Implementierung nicht in allen Details exakt umsetzen, was verschiedene Gründe haben kann. Bisweilen entpuppt sich auch eine andere Umsetzung als vorteilhafter. Die folgenden Abschnitte halten für jedes Softwaremodul die Abweichungen der Implementierung gegenüber den im Entwurf beschriebenen Strukturen fest. Des Weiteren enthalten sie die Gründe für diese Abweichungen.

3.1 Model

3.1.1 Ersetzen des Entwurfsmuster Erbauer durch eine Zuständigkeitskette

Das Paket „Model.BuildingBlockBuilder“ im Entwurf wurde durch das Paket „model.block“ ersetzt. Das dort verwendete Entwurfsmuster Erbauer erfüllte nicht die notwendige Anforderung, dass der Benutzter leicht eigene Versionen von Bausteinen in die Anwendung einfügen konnte. Darum wurde der Erbauer durch eine Zuständigkeitskette ersetzt. Hier gibt es keine Methode für jeden Baustein im Director, sondern es gibt nur eine Anzahl von Bearbeitern, die einen Block eines Types erstellen. Wenn also der Benutzter eine eigene Transformation erstellen will, kann er die .yaml Datei einer bereits vorhandenen Transformation kopieren und einige Parameter (außer Typ und subtyp) verändern. Die resultierende Transformation wird dann von der Anwendung als eine erkannt und kann dann auch dort verwendet werden. Dadurch entfallen alle folgenden Klassen des Entwurfs:

Builder

TransformationBuilder

RepresentationBuilder

XYRepresentationBuilder

TableRepresentationBuilder

SensorBuilder

VirtualSensorBuilder

PhysicalSensorBuilder

SnakeYamlParser

java.util.hashmap

sowie all diese öffentlichen Methoden in der Director Klasse:

createSensorFromYaml

constructTransformation

constructXYRepresentation

constructNTimeRepresentation

constructDS18B20TemperatureSensor

constructBMPx80PressureSensor

constructINA219CurrentAndVoltageSensor

constructMMA8451Accelerometer

constructTransformation

Statt dessen wurden folgende Klassen hinzugefügt:

GeneralBlockKvProcessor

KvProcessor

SensorKvProcessor

PhysicalSensorKvProcessor

VirtualSensorKvProcessor

RepresentationKvProcessor

TableRepresentationKvProcessor

XYRepresentationKvProcessor

TransformationKvProcessor

und die Methode constructBuildingBlock zum Director und zu jedem Bearbeitern die Methode processKvPair hinzugefügt. Dabei unterscheiden sich die Methoden der einzelnen Bearbeitern zwar nicht im Namen, aber in ihrer Funktion. Jeder Bearbeiter leitet entweder die Anfrage weiter oder erstellt einen Blocktyp und gibt ihn zurück.

3.2 GUI

3.3 Controller

3.4 Backend

3.5 Cache

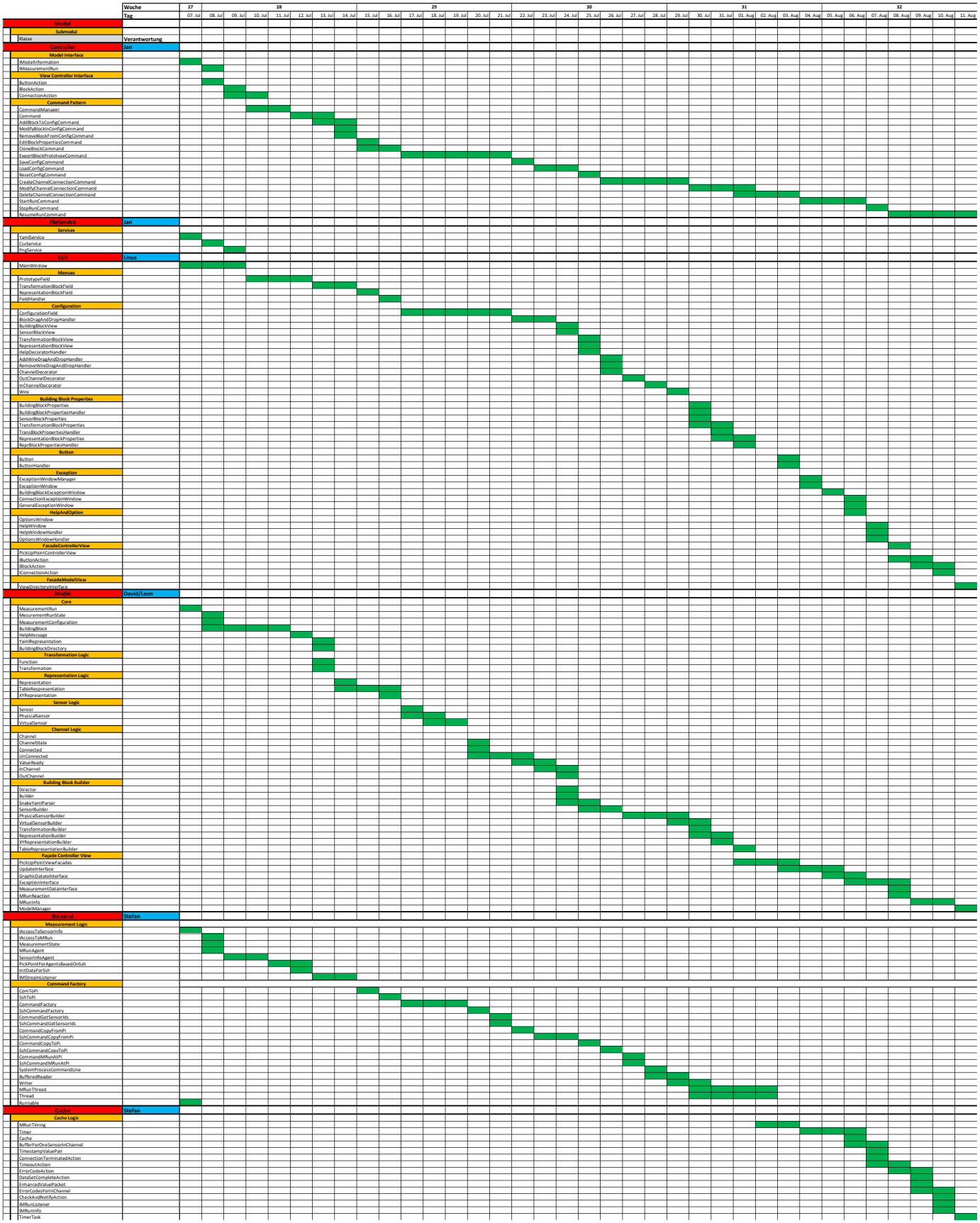
3.6 File-Service

4 Realer Implementierungsablauf

Dieser Abschnitt führt auf, inwieweit der tatsächliche zeitliche Implementierungsablauf vom geplanten Ablauf abgewichen ist, und beschreibt die Ursachen und Gründe für diese Abweichungen. Abhängigkeiten zwischen den Implementierungsschritten und kritische Pfade stehen hierbei besonders im Fokus.

Von Abweichungen betroffene Softwareelemente werden nicht im Einzelnen aufgeführt, sondern es werden lediglich in Bezug auf die Abweichungsgründe die Gruppen der betroffenen Softwareelemente benannt.

5 Anhang


Woche
27
28
29
30
31
32
Tag
07. Jul
08. Jul
09. Jul
10. Jul
11. Jul
12. Jul
13. Jul
14. Jul
15. Jul
16. Jul
17. Jul
18. Jul
19. Jul
20. Jul
21. Jul
22. Jul
23. Jul
24. Jul
25. Jul
26. Jul
27. Jul
28. Jul
29. Jul
30. Jul
31. Jul
01. Aug
02. Aug
03. Aug
04. Aug
05. Aug
06. Aug
07. Aug
08. Aug
09. Aug
10. Aug
11. Aug
Modul
Submodul
Klasse
Verantwortung
Controller
Jan
Model Interface
IModelInformation
IMeasurementRun
View Controller Interface
ButtonAction
BlockAction
ConnectionAction
Command Pattern
CommandManager
Command
AddBlockToConfigCommand
ModifyBlockInConfigCommand
RemoveBlockFromConfigCommand
EditBlockPropertiesCommand
CloneBlockCommand
ExportBlockPrototypeCommand
SaveConfigCommand
LoadConfigCommand
ResetConfigCommand
CreateChannelConnectionCommand
ModifyChannelConnectionCommand
DeleteChannelConnectionCommand
StartRunCommand
StopRunCommand
ResumeRunCommand
FileService
Jan
Services
YamlService
CsvService
PrjService
GUI
Linus
MainWindow
Menues
PrototypeField
TransformationBlockField
RepresentationBlockField
FieldHandler
Configuration
ConfigurationField
BlockDragAndDropHandler
BuildingBlockView
SensorBlockView
TransformationBlockView
RepresentationBlockView
HelpDecoratorHandler
AddWireDragAndDropHandler
RemoveWiresDragAndDropHandler
ChannelDecorator
OutChannelDecorator
InChannelDecorator
Wire
Building Block Properties
BuildingBlockProperties
BuildingBlockPropertiesHandler
SensorBlockProperties
TransformationBlockProperties
TransBlockProoertiesHandler
RepresentationBlockProperties
ReprBlockPropertiesHandler
Button
Button
ButtonHandler
Exception
ExceptionWindowManager
ExceptionWindow
BuildingBlockExceptionWindow
ConnectionExceptionWindow
GeneralExceptionWindow
HelpAndOption
OptionsWindow
HelpWindow
HelpWindowHandler
OptionsWindowHandler
FacadeControllerView
PickUpPointControlerView
IButtonAction
IBlockAction
IConnectionAction
FacadeModelView
ViewDirectoryInterface
Model
David/Leon
Core
MeasurementRun
MesurementRunState
MeasurementConfiguration
BuildingBlock
HelpMessage
YamlRepresentation
BuildingBlockDirectory
Transformation Logic
Function
Transformation
Representation Logic
Representation
TableRespresentation
XYRepresentation
Sensor Logic
Sensor
PhysicalSensor
VirtualSensor
Channel Logic
Channel
ChannelState
Connected
UnConnected
ValueReady
InChannel
OutChannel
Building Block Builder
Director
Builder
SnakeYamlParser
SensorBuilder
PhysicalSensorBuilder
VirtualSensorBuilder
TransformationBuilder
RepresentationBuilder
XYRepresentationBuilder
TableRepresentationBuilder
Façade Controller View
PickUpPointViewFacades
UpdateInterface
GraphicDataInterface
ExceptionInterface
MeasurementDataInterface
MRunReaction
MRunInfo
ModelManager
Backend
Stefan
Measurement Logic
IAccessToSensorInfo
IAccessToMRun
MeasurementState
MRunAgent
SensorInfoAgent
PickPointForAgentsBasedOnSsh
InitDataForSsh
IMStreamListener
Command Factory
ComToPi
SshToPi
CommandFactory
SshCommandFactory
CommandGetSensorIds
SshCommandGetSensorIds
CommandCopyFromPi
SshCommandCopyFromPi
CommandCopyToPi
SshCommandCopyToPi
CommandMRunAtPi
SshCommandMRunAtPi
SystemProcessCommandLine
BufferedReader
Writer
MRunThread
Thread
Runnable
Cache
Stefan
Cache Logic
MRunTiming
Timer
Cache
BufferForOneSensorInChannel
TimestampValuePair
ConnectionTerminatedAction
TimeoutAction
ErrorCodeAction
DataSetCompleteAction
EnhancedValuePacket
ErrorCodesForInChannel
CheckAndNotifyAction
IMRunListener
IMRunInfo
TimerTask

6 Glossar