

Moduldesign: MainWindow

Grafisches Eisenbahn Frontend (GEF)

Author:	Nils Neemann
Version:	0.2
Status:	nicht freigegeben / in Bearbeitung

Inhalt

1	Einleitung.....	3
2	Architektur.....	3
2.1	Allgemein.....	3
2.2	SerialInterface	4
2.3	Interpreter	4
2.4	ConfigWidget.....	5
3	Schnittstellen.....	5
3.1	Allgemein: Statusnachrichten der Module empfangen	5
3.2	Serielle Schnittstelle: Empfang von eingehenden Daten	6
3.3	Interpreter: Empfang von eingehenden (interpretierten) Daten.....	6
3.4	ConfigWidget: Konfiguration wurde geändert.....	7
3.5	Referenzdokumente.....	8

Änderungen

Version	Datum	Author	Änderungen
0.0	15.12.2010	Nils Nesemann	Erstellung des Dokuments
0.1	16.12.2010	Nils Nesemann	Überarbeiten der Modulschnittstellen
0.2	04.01.2011	Nils Nesemann	-Überarbeiten des Layouts -Ergänzen der Schnittstellenbeschreibungen -Einfügen von Sequenzdiagrammen

1 Einleitung

Das Hauptfenster(Main Window) übernimmt als zentrales Modul die Koordination aller Module und bildet die Basis für die Interaktion mit dem Benutzer. Die Kernfunktionen sind:

- Benutzereingaben behandeln
- Steuerung der Module (Initialisieren, Starten, Stoppen,...)
- Steuern des Datenflusses im System

2 Architektur

Die Funktionalität des Hauptfensters besteht aus verschiedenen Sequenzen die von den anderen Modulen angestoßen werden. Folgende Sequenzen sind im Hauptfenster zu implementieren.

2.1 Allgemein

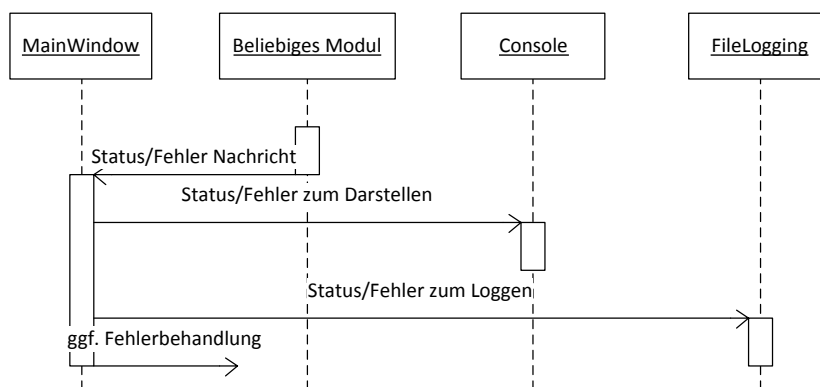


Abbildung 1: Sequenz zum Abarbeiten von eingehenden Statusmeldungen

2.2 SerialInterface

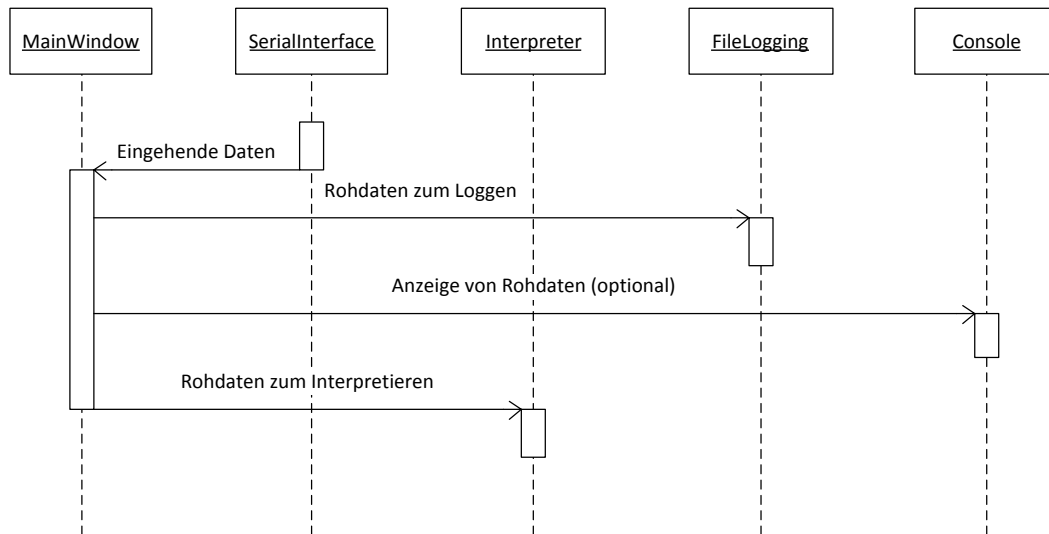


Abbildung 2: Sequenz zum Abarbeiten von eingehenden Daten

2.3 Interpreter

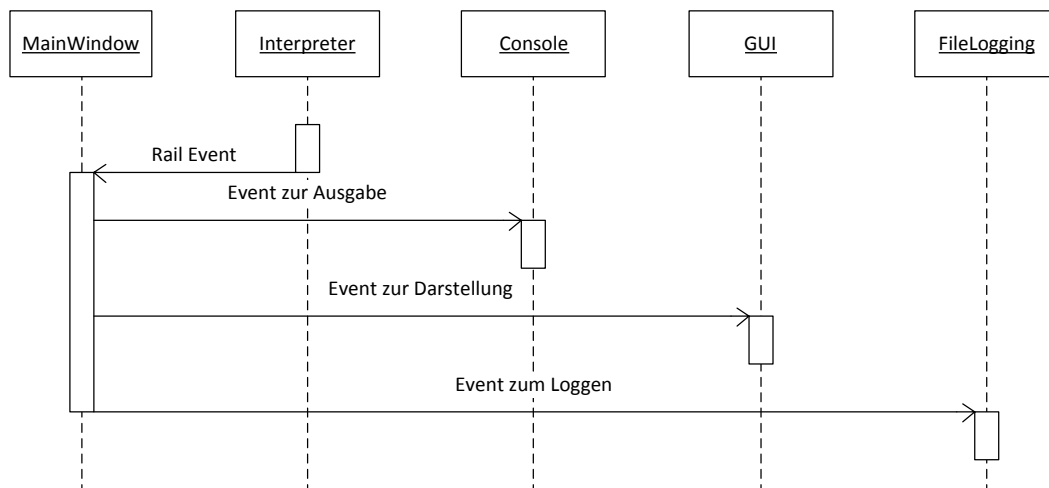


Abbildung 3: Sequenz zum Abarbeiten von Rail Events

2.4 ConfigWidget

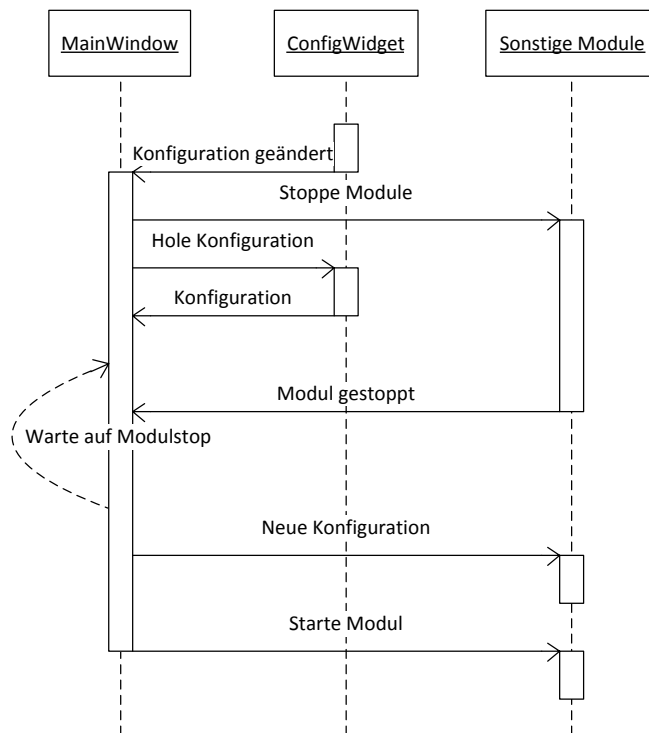


Abbildung 4: Sequenz bei Änderung der Konfiguration

3 Schnittstellen

Die vom Hauptfenster zur Verfügung gestellten Schnittstellen sind als QT-Slots zu implementieren.

3.1 Allgemein: Statusnachrichten der Module empfangen

Es wird eine Schnittstelle benötigt an welche alle Module ihre Statusmeldungen senden können. Diese Nachrichten müssen neben den Nutzdaten auch ihre Herkunft und einen Nachrichtentyp enthalten.

Tabelle 1: Quellen-Kodierung von Status-Nachrichten

Int-Wert	Quelle	Beschreibung
1	SOURCE_SERIAL	Statusmeldung von der Seriellen Schnittstelle
2	SOURCE_GUI	Statusmeldung vom GUI-Widget

3	SOURCE_CONFIG	Statusmeldung vom Config-Widget
4	SOURCE_SYSTEM	Statusmeldung vom Hauptfenster
5	SOURCE_INTERPRETER	Statusmeldung vom Interpreter
6	SOURCE_LOGGING	Statusmeldung vom Dateilog
7	SOURCE_CONSOLE	Statusmeldung von der Konsolen-Widget

Tabelle 2: Statuscodes

Int-Wert	Status	Bedeutung
-1	TYPE_ERROR	Fehlermeldung
1	TYPE_STATUS_LVL_1	Bestätigungen von Hauptfunktionen (z.B. Gestartet, Gestoppt, Gesendet, etc.)
2	TYPE_STATUS_LVL_2	Grobe interne Abläufe der Hauptfunktionen (z.B. Starte, Stoppe, Rufe unterfunktion XY auf, ...)
3	TYPE_STATUS_LVL_3	Detaillierte interne Abläufe
4	TYPE_STATUS_LVL_4	Meldungen von Unterfunktionen (zB Checksumme gebaut, GUI gezeichnet, etc.)

3.2 Serielle Schnittstelle: Empfang von eingehenden Daten

Die von der Seriellen Schnittstelle empfangenen Daten sind QByteArrays variabler Länge.

3.3 Interpreter: Empfang von eingehenden (interpretierten) Daten

Die interpretierten Daten ergeben ein das Schienennetz betreffendes Event (RailEvent) welches die folgenden Daten enthält:

Betreffender Objekttyp, Objektnummer, Ereignis, und Daten zum Ereignis

Tabelle 3: Objekte im Schienennetz

Int-Wert	Objekt	Beschreibung
1	RAIL_SENSOR	Sensor im Schienennetz
2	RAIL_COUPLER	Entkoppler
3	RAIL_SWITCH	Weiche
4	RAIL_TRAIN	Zug

Tabelle 4: Events im Schienennetz

Int-Wert	Event	Beschreibung
1	RAIL_EVENT_ACTIVATED	Ein Objekt im Schienennetz wurde aktiviert (Sensor überfahren, Weiche gestellt, ...)
2	RAIL_EVENT_SPEED	Die Geschwindigkeit eines Zuges wurde angepasst.

3.4 ConfigWidget: Konfiguration wurde geändert

Bei geänderter Konfiguration wird das Hauptfenster benachrichtigt und muss darauf reagieren. Es werden keine Übergabeparameter benötigt.

3.5 Referenzdokumente

Systemdesign