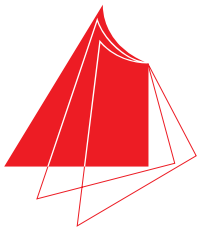

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Entwurf einer Architektur und eines Proof of Concept für ein echtzeitfähiges SCADA System mit Webfrontend

Bachelor-Thesis

zur Erlangung des akademischen Grades
Bachelor of Engineering

vorgelegt von

Florian Weber (44907)

12.11.2019

Erstprüfer: Prof. Dr.-Ing. Philipp Nenninger
Zweitprüfer: Prof. Dr. Stefan Ritter

Inhaltsverzeichnis

Abkürzungsverzeichnis

Kurzform Langform

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

1.1 Motivation

1.2 Zielsetzung

1.3 Gliederung

In der heutigen Informatik wird immer mehr auf grafische Benutzeroberflächen gesetzt, welche als Webapplikation im Browser lauffähig sind. In der Industrie ist dies leider noch nicht der Fall und die Benutzeroberfläche wird statisch geschrieben und kompiliert, was einige Einschränkungen mit sich bringt:

- Die Benutzeroberfläche ist statisch und lässt sich zur Laufzeit nur eingeschränkt an Daten anpassen. So kann man zum Beispiel ein interaktives Steuerelement zur Laufzeit nur sichtbar oder unsichtbar machen, eine dynamische Bindung an Daten ist nicht möglich.
- Die Benutzeroberfläche ist Plattform gebunden. Wenn zum Beispiel in WinCC eine Oberfläche erstellt wird, braucht man zwingend eine WinCC Runtime welche ausschließlich in Windows lauffähig ist.

Um die genannten Probleme zu lösen wird eine neuartige Architektur entworfen um Prozessdaten im Browser in Echtzeit anzuzeigen und zu verändern. Die folgenden Anforderungen werden an die Architektur gestellt:

- Die Datenrate der Webapplikation bei vertretbarem Aufwand, soll so klein wie möglich sein. Dies ermöglicht unteranderem die Nutzung des Systems im einem Umfeld mit geringer verfügbarer Bandbreite zur Steuerungsebene.
- Das Framework soll Steuerelemente die eine Eingabe durch den Nutzer zulassen, sowie Steuerelemente die eine Darstellung eines Prozesswerts ermöglichen.
- Die Webapplikation selbst soll so modular sein, dass man zur Laufzeit Steuerelemente hinzufügen und entfernen kann, ohne dass das System offline geht.
- Die Prozessdaten sollen nicht, wie aktuell bei vielen Webapplikationen üblich, durch Polling synchronisiert werden, sondern die Weboberfläche soll auf Datenänderungen des Prozesses in Echtzeit (bei statischem Routing im Netzwerk) reagieren. Dasselbe gilt für die andere Richtung.

- Die Architektur soll herstellerunabhängige Schnittstellen zur Integration in ein vorhandenes System bereitstellen.
- Die Weboberfläche soll eine feste Auflösung haben und muss nicht auf Änderungen des Viewports reagieren. Ausnahmen bilden hierbei Darstellungen die dies, durch ihre einfache Gestalt, erlauben wie zum Beispiel:
  - Tabellen
  - Buttons
  - Labels
  - Einzelne Grafiken deren Position für die Darstellung nicht relevant ist

Der Beweis der Realisierbarkeit soll durch eine Beispielimplementierung erbracht werden. Dabei soll je ein Eingabeelement (z.B. Button), ein Ausgabeelement (z.B. Label), sowie ein Ein-/Ausgabe Element (zum Beispiel ein Textfeld) implementiert werden.

2 Theoretische Grundlagen

2.1 SCADA

2.2 Verteilte Systeme

2.2.1 Synchrones Datenmodell

2.2.2 Asynchrones Datenmodell

2.3 Security

2.3.1 Authentifizierung

2.3.2 Verschlüsselung

2.4 Safety

2.5 Websocket

2.5.1 TCP

2.6 SQL

2.7 Echtzeitfähigkeit

2.8 OPCUA

3 Aktueller Stand der Technik

lsg vetablierter hersteller

3.1 Gui zeug auto

3.2 transport zeug

3.3 datenmanagement

4 Systementwurf

4.1 Architektur

4.2 Use-Cases

4.2.1 Backend

4.2.2 Frontend

5 Umsetzung des Proof of Concept

5.1 backend

5.1.1 HIER KLASSEN

5.2 Frontend

6 Zusammenfassung und Ausblick

1

[1] John Doe. „How do I get this to work?“ In: *TeX Monthly* 99 (1234), S. 1–10.

Literatur

Doe, John. „How do I get this to work?“ In: *TeX Monthly* 99 (1234), S. 1–10.