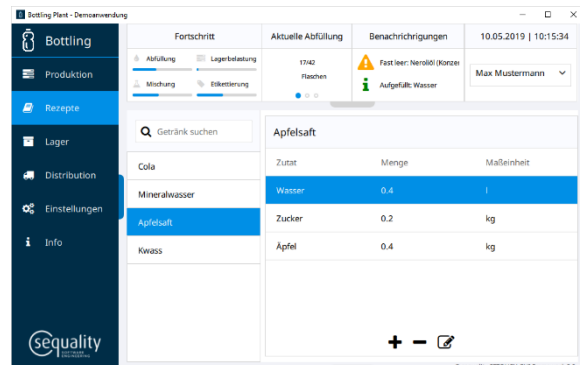


QML without C++: A new architecture to write QtQuick UIs

Flüssig animierte graphische Benutzeroberflächen, Unterstützung für OpenGL und eine einfache Entwicklung von Touch-Display Controls führten vor gut 10 Jahren zur Entwicklung des Moduls QtQuick im Qt Framework, das mit der deklarativen Programmiersprache QML genutzt werden kann um interaktive graphische Benutzeroberflächen umzusetzen. QML fokussiert dabei auf die zustandsbasierte Darstellung der Elemente und



bietet 2 Möglichkeiten an, um prozedurale Logik auszuführen: Einerseits kann (1) Javascript direkt in QML Komponenten genutzt werden um einfache Logiken auszuführen, andererseits existiert eine (2) C++ Schnittstelle um Qt Objekte in den Kontext der QML Anwendung zu übergeben. Insbesondere für die Übergabe von größeren Datenmengen (z.B. von einer Datenbank oder von JSON Datenströmen) war man auf die Qt Model-View Architektur angewiesen, die die Übergabe von Qt C++ Models nach QML definiert.

Die Master-Arbeit hat zum Ziel, eine neue Architektur für die Umsetzung von QtQuick Anwendungen vorzuschlagen, welche möglichst ohne der Verwendung von C++ auskommt. Dabei soll der aktuelle Qt Model-View Ansatz (welcher vorgibt die Datenmodelle in C++ zu implementieren) durch einen neuen QML Model-View Ansatz ersetzt werden, welcher im Aufbau eher einer modernen HTML5 Single-Page-Web-Applikation ähnelt: Sämtliche Datenmodelle werden hier via JSON-Nachrichten über einen Websocket geschickt. Um dies im Qt Framework bequem zu unterstützen wird es notwendig sein, zusätzliche Qt Komponenten bereitzustellen, um den neuen QML Model-View Ansatz optimal zu unterstützen.

Aufgabenstellung:

- Untersuchung der aktuellen Möglichkeiten QtQuick-Anwendungen ohne C++ umzusetzen
- Vorschlag einer neuen QML Model-View Architektur, die auf die Verwendung von Datenströmen von Websockets (JSON) optimiert ist und bei der Verwendung ohne C++ auskommt.
- Umsetzung von zusätzlichen Qt Komponenten um diese neue Architektur zu unterstützen.
- Demonstration/Evaluierung der neuen QML Model-View Architektur anhand eines Prototyps.

Technologie-Umfeld:

- C++, Qt, QML, Websockets
- Qt Model View Architektur

Kontakt:

Stefan Larndorfer, Stefan.Larndorfer@sequality.at
 sequality software engineering, Softwarepark 26, A-4232 Hagenberg
www.sequality.at