

Next Generation Peripherals Data

Themenbereiche:	Softwareentwicklung
Studierende:	Remo Grab
Betreuungsperson:	Dr. Martin Bättig
Experte:	Dominik Witschard
Auftraggebende:	Siemens Schweiz AG
Keywords:	Fire Detector, Metadata, VCS, Python, Graph

1. Aufgabenstellung

Die Fire Systems Abteilung der Siemens in Zug entwickelt und getestet Brandmelder. Die Brandmelder ermitteln den physischen Zustand eines Raumes mittels Sensoren und berechnen Gefahrenlevel und kommunizieren bei Brandverdacht mit einer zentralen Steuerungseinheit. Die Kommunikation zwischen Steuerungseinheit und den Brandmeldern findet dabei über ein in der Siemens entwickeltes Protokoll statt, welches auf Stabilität ausgerichtet ist, was jedoch zu Kosten der Datenrate kommt. Aufgrund der limitierten Datenrate kommunizieren die Melder in einem platzsparenden Binärformat. Die binären Nachrichten enthalten einen Nachrichtentyp, welcher dem Empfänger mitteilt, wie die Nachricht zu interpretieren ist. Ein Teil der Nachrichtentypen sind fest definiert und für alle Gerätetypen gleich und ein weiterer Teil ist abhängig vom Gerätetyp. Um den Inhalt dieser Nachrichten zu interpretieren, werden zusätzliche Metadaten benötigt, welche in einem existierenden Projekt, den platform-p2-data, unterhalten werden. Platform-p2-data ist ein Projekt, welches über mehrere Jahre angewachsen ist und an seine technischen Grenzen stösst. Unter anderem, dass die Daten nur pro Gerätetyp, nicht aber für verschiedene Versionen eines Gerätetyps unterhalten werden, was das Arbeiten mit älteren Geräten erschwert. Das Ziel dieses Projektes war es einen Nachfolger für die platform-p2-data zu kreieren.

2. Lösungskonzept

Es wurde ein System entwickelt, welches es erlaubt die Daten als Graph zu beschreiben. Bei der Erstellung der Metadaten einer neuen Geräteiteration wird auf die alten Metadaten verwiesen und somit können die neuen Metadaten rein anhand der Änderungen zur Vorgängerversion beschrieben werden. Die Beschreibung der Daten in diesem Format dient jedoch nur als Quelle, welche ein Builder in JSON-Files exportiert. Dieser Export wird durchgeführt damit der Versionierungsaufbau nicht zu Laufzeiten aufgelöst werden muss und somit Laufzeitprobleme vermieden werden können. Ein einfaches Beispiel ist in Abbildung 1 zu sehen. In der oberen Reihe sind die verschiedenen Versionen der Daten zu sehen, welche jeweils auf ihren Vorgänger verweisen und die Änderungen als Attribute beinhalten. In der unteren Reihe die gebauten Metadaten zu sehen, bei welchen der Versionierungsaufbau aufgelöst wurde.

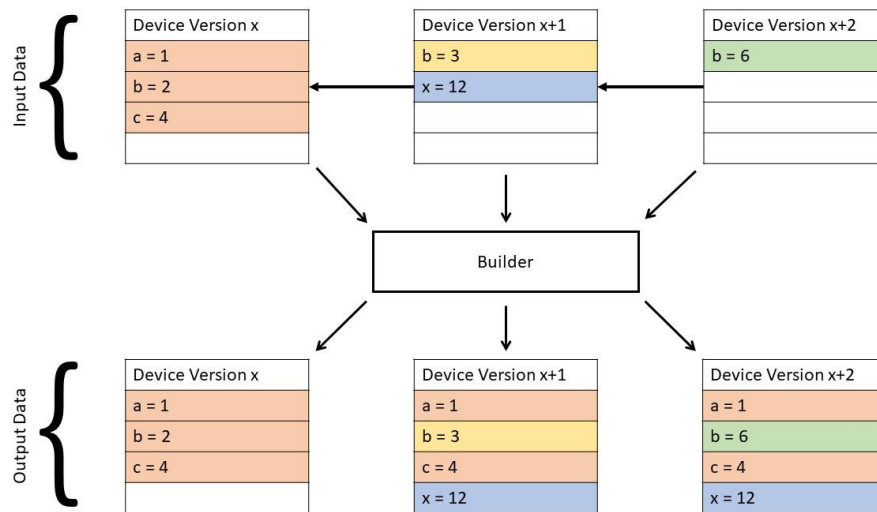


Abbildung 1: Verlinkte Input Daten werden von einem Builder gebaut

3. Spezielle Herausforderungen

Folgende Probleme wurden im Laufe des Projekts analysiert und gelöst:

- Die Brandmelder, für welche die Metadaten gehalten werden, verfügen über zwei unterschiedliche Versionsnummern, die FS (Functional State) und die SWREV (Softwarerevision). Das Ziel war es, dass eine API aus mehreren verfügbaren Versionen der Metadaten eine Best-Guess Version anhand des Gerätetyps und optional einer oder beider Nummern evaluiert. Eine zusätzliche Schwierigkeit für diese Aufgabe war, dass bestimmte Versionsnummern für In-Development Geräte reserviert sind und diese dürfen nicht als Best-Guess evaluiert werden.
- Die Daten sollen offline verfügbar sein. Da ein Teil der Testlabore und Feldtechniker keinen Internetzugriff haben, musste sichergestellt werden, dass es möglich ist die Daten im Vorfeld bereit zu stellen.
- Da es möglich ist, dass auf einem Computer mehrere Programme laufen, welche unterschiedliche Versionen der Peripherals-Data benötigen, musste sichergestellt werden, dass mehrere Versionen der Metadaten auf einem Computer verfügbar sein können und dass die API, welche diese Daten lädt, dazu fähig ist, aus mehreren Datenreleases die Korrekten auszuwählen.

4. Ergebnisse

Die erzeugten Ergebnisse der Arbeit beinhalten:

- **Peripherals Data Bibliothek:** Ein Python Package, welches über die Artifactory der Siemens mittels PIP (Python Package Manager) installiert werden kann. Das Package ist dazu fähig die Metadaten eines Gerätes nach dem Gerätetyp, FS und SWREV zu liefern.
- **Definition eines Datenformats:** Definition eines Datenformats zur Haltung der Daten in einem Quellformat und der Implementierung eines Builders, welcher die Daten in JSON-Files exportiert und in einem Zip-File verpackt. Somit können die Daten auch problemlos heruntergeladen und offline verwendet werden.
- **Integration des erstellten Packages in eine existierende Software:** Das Package wurde in ein existierendes Rohdatenloggingprogramm integriert. Die Rohdaten werden vom Brandmelder als ein grosser Block verschickt, das Loggingprogramm nimmt diesen Block entgegen und zersetzt ihn mit Hilfe der Metadaten in seine Einzelteile. Zuvor war es nicht möglich die Rohdaten von zwei Geräte vom gleichen Typ mit unterschiedlichen Rohdatenversionen auszulesen. Mit dem neuen System ist dies problemlos möglich, da die Metadaten nicht nur nach Gerätetyp, sondern auch Geräteversion verfügbar sind.

5. Ausblick

Beim Projekt lag der Fokus darauf einen Prototyp zu entwickeln, welcher mit der Rohdateninterpretation einen ersten Use-Case abdecken kann. Als nächstes erfolgt die Überführung der Daten von existierenden Geräten aus dem platform-p2-data Projekt in das neue Format. Weiter werden Bibliotheken so angepasst, dass sie die neuen Peripherals-Daten verwenden und somit das platform-p2-data Projekt vollständig abgelöst werden kann.