

DataInspector - Erweiterung einer abteilungsspezifischen C++ Library

Praxisarbeit

über die Praxisphase des dritten Studienjahrs

an der Fakultät für Technik
im Studiengang Informationstechnik

an der Duale Hochschule Baden-Württemberg (DHBW) Ravensburg
Campus Friedrichshafen

von
Nico Bayer

Frau Schmidt Mail nachfragen

Bearbeitungszeitraum:	11 Wochen, KW 2-13
Matrikelnummer, Kurs:	3056312, TIT20
Dualer Partner:	Hensoldt Sensors GmbH
Betreuer des Dualen Partners:	Dipl. Phys. Ulrich Hierl

Selbstständigkeitserklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017.

Ich versichere hiermit, dass ich meine Bachelorarbeit (bzw. Projektarbeit oder Studienarbeit bzw. Hausarbeit) mit dem Thema:

DataInspector - Erweiterung einer abteilungsspezifischen C++ Library

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

HEYD4

Abteilung, Unterschrift

Abstract

Zusammenfassung

Schlüsselwörter

Abstract

Keywords

Inhaltsverzeichnis

Abkürzungsverzeichnis	V
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VI
1 Einleitung	1
2 Aufgabenstellung	2
3 Stand der Technik	4
3.1 testing	4
4 Lösungsansatz und Implementierung	5
5 Testen der Software	6
6 Ausblick	7
Referenzen	A
A Anhang	A

Abkürzungsverzeichnis

DHBW Duale Hochschule Baden-Württemberg	I
--	----------

Abbildungsverzeichnis

2.0.1 Das Kontext-Diagramm der Arbeit mit Hervorhebung der zu bearbeitenden Teile. 3

Tabellenverzeichnis

Listingsverzeichnis

1 Einleitung

Diese Arbeit wurde in der Abteilung HEYD4, im Team Electronic Warfare Data Generation System, der Hensoldt Sensors GmbH verfasst, welches Ground-Support Softwarelösungen für das „Defensive-Aid“ System des Eurofighters entwickelt. Die Firma ging 2017 aus dem Verkauf der Sensor- sowie optoelektronischen Sparten von Airbus Defence and Space hervor und widmet sich hauptsächlich der Produktion von Lösungen für die Verteidigungsbranche. Eine große Produktgruppe der Firma sind Radare.

Als Electronic Warfare (EW) bezeichnet man jede Aktion, die die Nutzung des elektromagnetischen Spektrums oder gerichteter Energie beinhaltet, um das Spektrum zu kontrollieren, einen Feind anzugreifen oder feindliche Angriffe zu verhindern.

Der seit 2003 in Serie produzierte Eurofighter Typhoon besitzt EW-Hardware Komponenten für verschiedene Zwecke. Da nicht für jede Mission sämtliche Hardware, beziehungsweise in unterschiedlicher Art und Weise, benötigt wird, muss diese vor Beginn einer Mission konfiguriert werden.

Die Konfiguration erfolgt vom Eurofighter Ground Support in Form einer binären Datei, welche auf die EW-Hardware des Luftfahrzeugs geladen wird. Da eine Binärdatei von Menschen nur mühsam erstellt werden kann, stellt das EWDGS für das Eurofighter Ground Support Team eine grafische Oberfläche, auch genannt Editor, zur Verfügung, mit deren Hilfe die Datei parametrisiert und erstellt werden kann.

Damit die Konsistenz der Daten gewährleistet wird, existiert eine Erweiterung für den Editor in Form einer C++ Bibliothek, der sogenannte „DataInspector“, welcher in seiner momentanen Form nur die Konsistenz von Daten innerhalb einer SQL Datenbank, anhand eines projektspezifischen Regelsatzes überprüfen kann. An diesem Punkt setzt die Projektarbeit an.

2 Aufgabenstellung

Seit 2020 entwickelt Hensoldt am EuroDASS Praetorian, welches als Selbstschutzsystem des Eurofighter Typhoon im Luftraum eingesetzt wird. Hensoldt unterstützt ebenfalls das Eurofighter Ground Support Team, welches einen In-Service-Support gemäß den hohen operationellen Anforderungen des Systems leistet, um somit die Leistungsfähigkeit und verlässliche Verfügbarkeit der Luftfahrzeuge sicherzustellen.

Die Electronical Warfare Komponenten des Systems werden fast ausschließlich durch Software gesteuert, weshalb eine Konfiguration des Selbstschutzsystems durch eine binäre Datei vorgenommen wird, welche auf das System aufgespielt werden kann.

Diese binäre Datei wird mit Hilfe des EW Generier System am Boden vor dem Abflug generiert. Das Generiersystem, auch Editor genannt, wird von Hensoldt entwickelt. Die bereits erwähnte binäre Datei zur Konfiguration der EW-Hardware muss gewissen durch die Hardware vorgegebenen Einschränkungen entsprechen. Mit sogenannten Consistency-Check Regeln wird überprüft, dass die vom Benutzer eingegebenen Daten diesen Einschränkungen gehorchen. Die Softwarekomponente, welche für die Prüfung der Einhaltung dieser Regeln verantwortlich ist, wird „DataInspector“ genannt. Der DataInspector wird als Erweiterung des Generier Systems in Form einer C++ Bibliothek eingebunden. Die bisherigen Versionen der Software beziehen alle zu überprüfenden Daten aus einer SQL-Datenbank, welche vor dem Erstellen der binären Konfigurationsdatei mit den CC-Regeln in Form einer SQL-Abfrage auf ihre Konsistenz überprüft werden.

Das EWDGS hat für den aktuellen Release der Software eine neue Version entwickelt, welche die Konfigurationsdaten nicht in einer SQL-Datenbank, sondern in XML-Files hält. Da die SQL-Abfrage für die Daten im XML-Format nicht möglich ist, muss die Software erweitert werden, sodass der „DataInspector“ auch für die XML-Editoren verwendet werden kann

Ziel der Arbeit Erweitern der C++ Bibliothek „DataInspector“, damit sie auch Daten aus XML-Dateien auf ihre Konsistenz gegen die projektspezifischen Richtlinien prüfen kann.

In den nachfolgenden beiden Diagrammen werden die Grenzen des Projekts aufgezeigt. Ein Blackbox-Diagramm veranschaulicht Ein- und Ausgaben. Die „Systemgrenzen“ der T3000 grenzen ein, dass nur der „DataInspector“, Teil dieser Praxisarbeit ist.

Für den Zugriff auf die XML-Dateien gibt es bereits eine existierende verallgemeinerte Datenzugriffsschicht in C++, welche in den „DataInspector“ eingebaut werden soll. Es muss ein Weg gefunden werden die CC-Regeln auf die Daten der Datenzugriffsschicht anzuwenden.

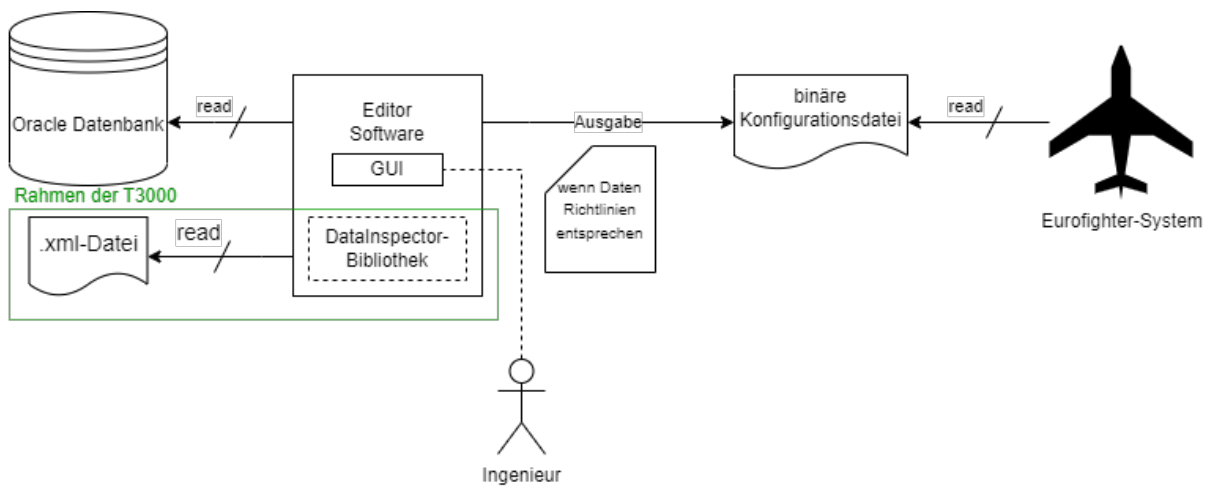


Abb. 2.0.1: Das Kontext-Diagramm der Arbeit mit Hervorhebung der zu bearbeitenden Teile.

3 Stand der Technik

In diesem Kapitel werden fundamentale Konzepte und benutzte Elemente erklärt, welche innerhalb der Arbeit erwähnt werden.

3.1 testing

4 Lösungsansatz und Implementierung

5 Testen der Software

In diesem Kapitel werden die Prinzipien erklärt, nach denen die Software getestet wurden.

6 Ausblick

A Anhang