

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
2.0.2 Das Use-Case Diagramm des Systems. Hier sind die wichtigsten Benutzer- Interaktionen in einer Grafik veranschaulicht.	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, welche zum größten Teil von der Verteidigungsbranche verwendet werden. Eine große Produktgruppe der Firma sind Radare.

Der seit 2003 in Serie produzierte Eurofighter Typhoon ist ein sicherheitskritisches System, was bedeutet, dass ein fehlerhaftes Verhalten der Software oder ein Ausfall des Systems zu Verlust von Menschenleben und Sachgütern von erheblichem Wert führen kann, weswegen ein korrektes Ergebnis vorliegen muss.

Die Software, welche sich innerhalb des Eurofighters befindet, muss für ihren jeweiligen Einsatzzweck angepasst werden. Hierfür ist das Eurofighter Ground Support Team zuständig, welches eine Parametrisierung der Eurofighter Software vornimmt.

Das Team EWDGS der Firma Hensoldt stellt für die Parametrisierung der Anwendung eine grafische Oberfläche zur Verfügung, mit dessen Hilfe ein Binärfile erzeugt werden kann. Aufgrund der sicherheitskritischen Natur des Gesamtsystems muss eine Konsistenz der Daten gewährleistet werden, weshalb es für die Editoren eine Erweiterung in Form einer C++ Bibliothek gibt, den sogenannten „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 Eurofighter Radar MK1, welches als mobiles Sensorsystem 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 Funktionalitäten des Flugzeuges werden fast ausschließlich durch Software gesteuert, weshalb eine Konfiguration des Fliegers durch eine binäre Datei vorgenommen wird, welche auf das System des Eurofighters aufgespielt werden kann.

Die Parametrisierung der Software, welche sich an Bord des Luftfahrzeuges befindet wird durch das Eurofighter Ground Support Team, über eine grafische Oberfläche, der von Hensoldt zur Verfügung gestellten Software, vorgenommen.

Die bereits erwähnte binäre Datei zur Konfiguration des mobilen Systems muss gewissen projektspezifischen Regeln entsprechen, welche Consistency-Check Richtlinien genannt werden. Die Anwendung, welche für die Prüfung der Einhaltung dieser Richtlinien verantwortlich ist, wird „DataInspector“ genannt. Der DataInspector wird als Erweiterung der Parametrisierungs-Software in Form einer C++ Bibliothek eingebunden. Die bisherigen Versionen der Software beziehen alle ihre Daten aus einer SQL-Datenbank, welche vor dem Erstellen der binären Konfigurationsdatei mit den CC-Richtlinien in Form einer SQL-Abfrage auf ihre Konsistenz überprüft werden.

Das EWDGS hat für den aktuellen Release der Software eine neue Version entwickelt, welche ihre Daten nicht aus einer SQL-Datenbank, sondern aus XML-Files holt. 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.

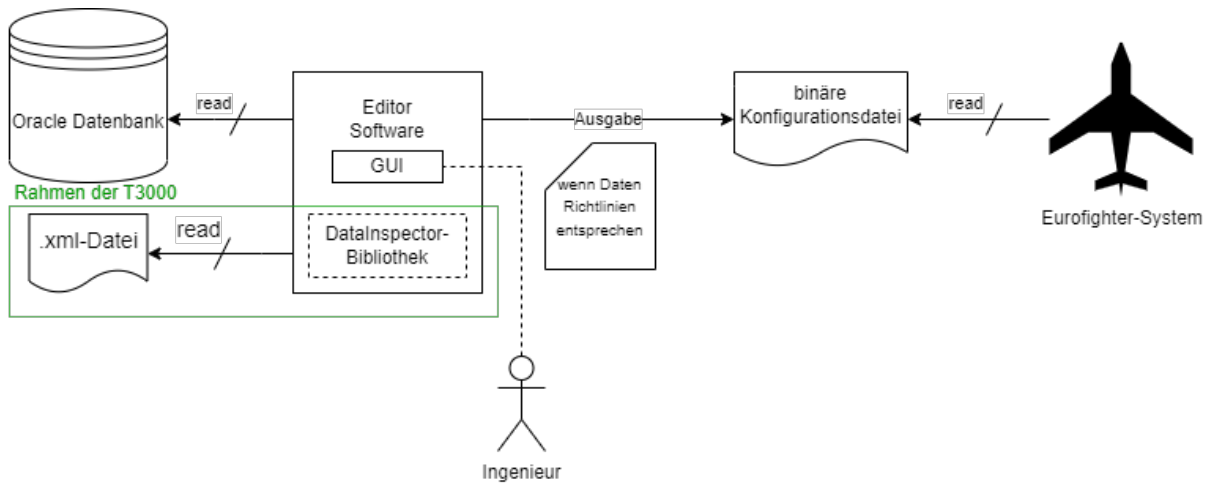


Abb. 2.0.1: Das Kontext-Diagramm der Arbeit mit Hervorhebung der zu bearbeitenden Teile.

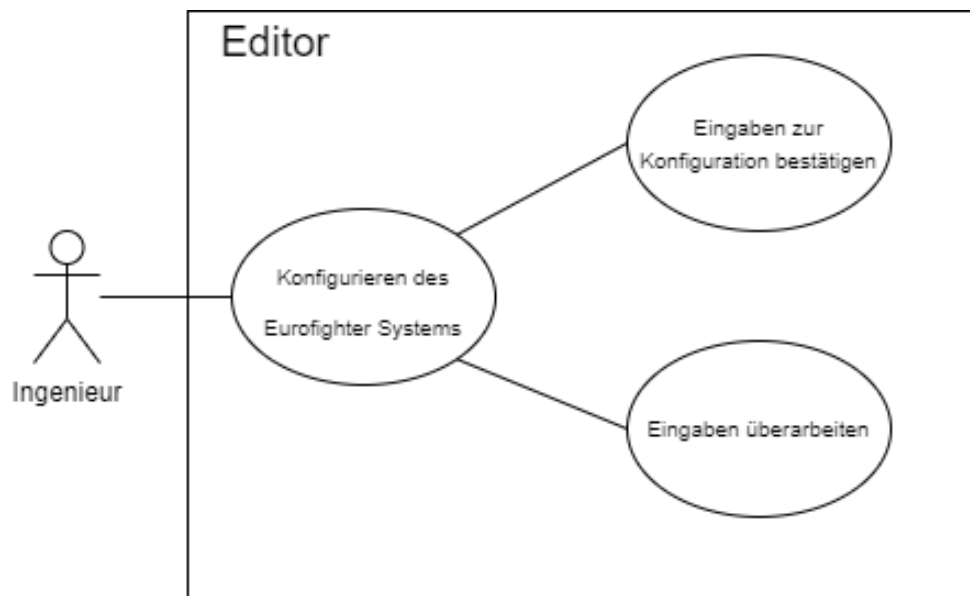


Abb. 2.0.2: Das Use-Case Diagramm des Systems. Hier sind die wichtigsten Benutzer-Interaktionen in einer Grafik veranschaulicht.

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