Acronym	Messer
Project	**ADS-B Message Server**
Doctype	**Requirements**
Author	**Hans-Gerhard Gross, Kai Warendorf**
Contact	hggross@hs-esslingen.de; Kai.Warendorf@hs-esslingen.de
Client	Esslingen University
Contact	Faculty of Information Technology
Version	1.0
Date	November 12, 2014

Contents

Chapter 1

Project Drivers

1.1 Purpose of the Project

1.1.1 Vision Statement

This project aims at developing a server that provides ADS-B messages locally in a Java application.

1.1.2 Project Outcomes

The Java application reads ADS-B sentences.

The Java application transforms each sentence into its respective message type.

The Java application prints a string representation of each message onto the screen.

1.1.3 Learning Objectives

After having completed this project, as student, you can ...

- develop and integrate Java classes and interfaces.
- apply the Java extension mechanism.
- perform advanced String transformation operations in Java.
- use Java's public inner classes.
- apply the Java observation/observable pattern.

1.2 Stakeholders

1.2.1 Project Team

Various members and roles.

1.2.2 Product Users

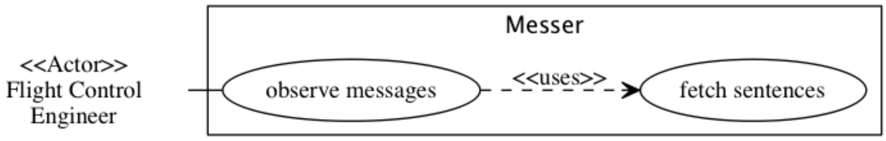
Local Flight Control Engineer, User. Priority: **Key User**.

Chapter 2

Functional Requirements

2.1 Data Model and Data Dictionary

2.1.1 Use Case Diagram



2.2 Messer Functional Requirements

Messer.F.10 Observe ADS-B Messages essential

Feature In order to get an overview of the local flight traffic, as I flight control engineer, I want to be able to observe each incoming ADS-B message.

Messer.F.20 Fetch Messages essential

Feature In order to provide ADS-B messages locally, the system shall fetch the corresponding sentences from the following web service:

`http://flugmon-it.hs-esslingen.de/subscribe/ads.sentence`

Feature In order to integrate seamlessly with other OS operations, the web service address shall be provided as input parameter upon application start.

Chapter 3

Non-Functional Requirements

3.1 Look and Feel Requirements

Messer.NF.10 Text Output per ADS-B message essential

Feature The system shall display each ADS-B message received in the following form (example):

```
484B91 Airborne Position Message
       Type:    12
       Alti:    30550
       Latlon:  15530 : 129595
       Format:  even

4B1621 Airborne Velocity Message
       Speed:   442
       Headng:  3
       Vertic:  -64

4B1621 Aircraft Identification and Category Message
       Ident:   SWR177Y
       Categ:   0

3C6424 Other Message
       Type:    29
```

The hexadecimal code represents the ICAO of the originator.

3.2 Implementation-Specific Requirements

3.2.1 Process

Messer.NF.50 Test Driven Development **essential**

In order to ascertain sufficient testing of the product, the implementation must be carried out following a test-driven development approach.

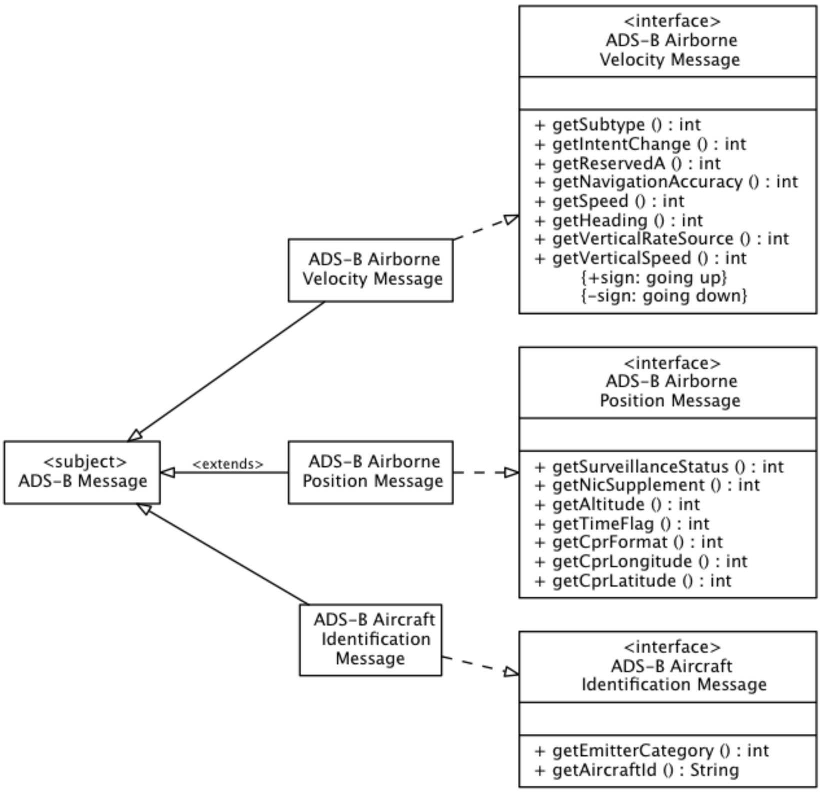
3.2.2 Architecture

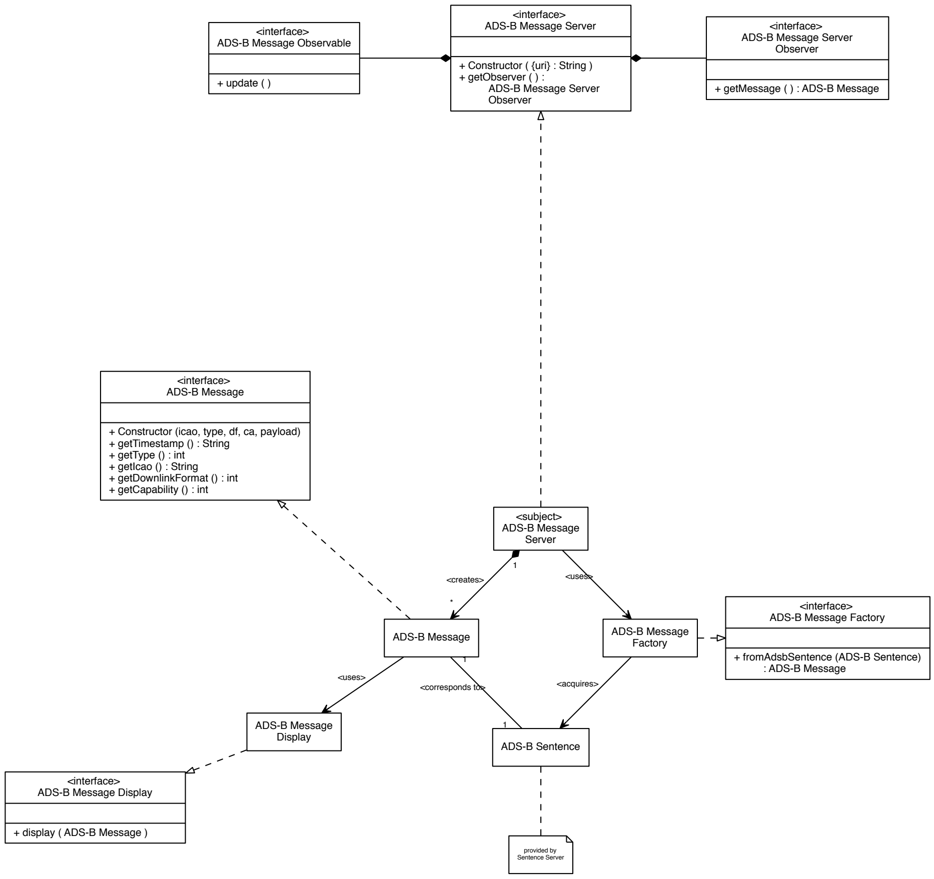
Messer.NF.60 Implementation of Messer **essential**

Feature In order to serve several clients at the same time in terms of a publish/subscribe architecture, the module *Messer*, i.e. the ADS-B Message Server, must be realized following the Observer/Observable architectural pattern.

Messer.NF.65 Use of Classes and Interfaces **essential**

Feature The organization of the system implementation shall reflect the classes and interfaces shown in the following class diagrams:





3.3 Maintainability Requirements

Messer.NF.70 Documentation essential

In order to ascertain high understandability, the source code must be self-explanatory.

Messer.NF.80 Cohesion and Coupling essential

In order to support high maintainability, the modules of the system must be realized with high-cohesion and low coupling.

Messer.NF.90 OO Design Principles essential

In order to support high maintainability, the other well-known principles of good object-oriented design must also be applied.