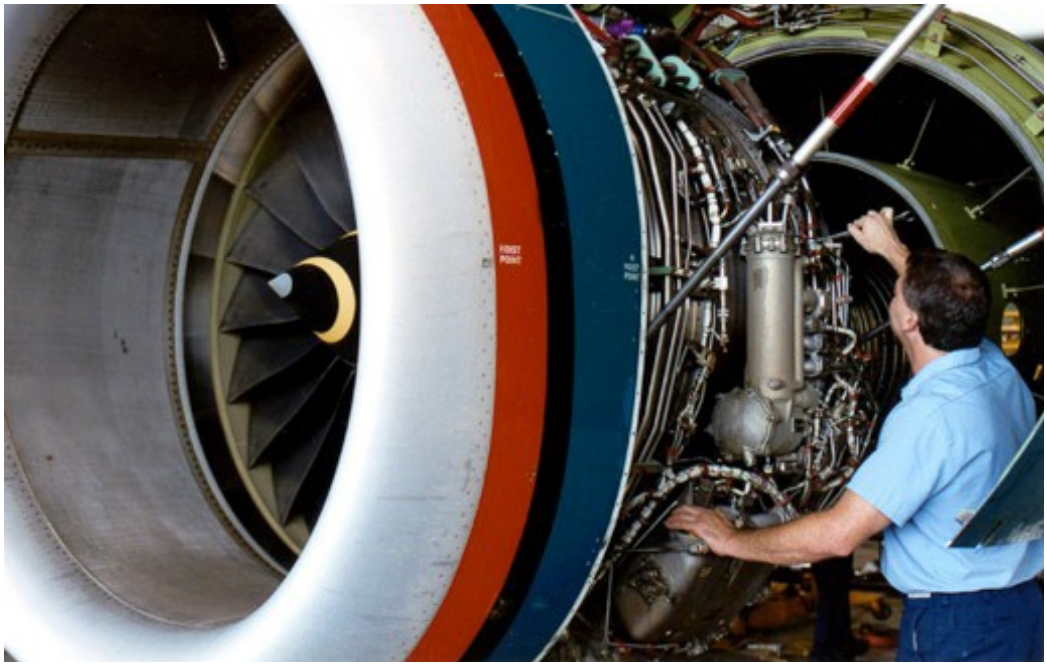


iARMED

Aircraft Maintenance System for iPhone

Chair for Applied Software Engineering
Prof. Bernd Brügge, Ph. D.
Dipl. Inf. Univ. Michael Nagel
Michael Huber

Summer Term 2010

Team
- Taha Dhiaeddine **A**mdouni
- Sonila **D**obi
- Benedikt **E**ngeser
- Steffen **S**trobel
- Marcel **S**tuht
- Benjamin **U**llrich

INDEX

I. Project Management Document

1. Introduction

Research in aircraft inspection and maintenance has revealed the criticality of human inspection performance in improving aviation safety.
Training Line Maintainers has been identified as the primary intervention strategy in improving the quality and reliability of aircraft inspection performance but the need of Experts remains.
Since Experts cannot be all around the world at the same time to coach the Line Maintainers, it is clear that we need to provide aircraft inspectors with tools to keep them in close contact with experienced staff and with a database recording the history of aircrafts.
In response to this need, a high fidelity wireless maintenance system for iPhone should be developed.

2. Purpose of the system

The application is a demo for an aircraft maintenance tool which serves for communication, controlling, assistance and reparation in distance. This software is of large dimension and integrates different components. The way it works, is similar to the work the Line Maintainers and Experts do in the airports around the world. Basically it does the work of routine After-Flight Inspection, planning for reparation, doing the actual reparation through given instructions, maintenance and job planning. All this is done electronically with the help of this tool which is being developed. Through video streaming the Expert does not have to leave his office but can do his work in distance through the use of wireless devices such as an iPhone.
The LM can record everything and save it for later analysis and can also be instructed from Experts just by using a phone.

The software is user friendly with basic and enough functions. It has a step by step intuitive user interface. It solves the daily tasks related to the maintenance in the airport, and now the distance maintenance is made even easier.

The information is saved in different data storage servers and they are all connected together at the same time to offer all the services. There is one server which will save the aircrafts models, another one which will save the attachments and another one which will ensure the streaming. All these are connected together with some URLs and the identification of the elements can be done with their IDs.

The workload data is changeable depending on how many problems occur per day.
The system should make the work easier for the users since they can access everything by clicking a “button”, therefore this software allows increasing the work and easing the job.

The input of the program is aircraft models and parts with attachments such as pictures, videos, and notes made from the LMs and the Experts.

The output consists of different types of media such as voices, videos, pictures, text and is accessible at any time.

3. Purpose of the documentation

The purpose of this documentation is to give information for all the developing stages of the software and present the work for a successful realisation of it. Here are explained the technical steps needed to build the system. This description might be used for future development in case similar products are going to be developed.
A detailed documentation makes it easier identifying parts that can be reused. This is the main reason for writing this documentation. Even though we might not want to admit it, it could be that at some point this software might exhibit some anomaly not foreseen from us. Consulting this documentation might help us solve the problem. This justifies documenting from the beginning of the development until delivery as a maintenance tool for future reference.

4. Project team and allocation of tasks

The thumb rule “there is no I in a team”.

The software team is composed from a supervisor, a coach and the programming members. The supervisor’s role was to identify possible problems during the process and provide with part of the implementation of the system, mostly the server side development.
The coach was side by side with the rest of the team dealing with the administrative aspect such as checking the status of the project, pondering the different proposed solutions, setting milestones, etc.
The other members had different tasks almost each of them.
Steffen had the task related to VoIP, Marcel the Graphical User Interface, Benjamin the WebDAV connector implementation, Benedikt the HTTP connector implementation and Sonila and Taha the software engineering modelling and documentation.
All the team members had to collaborate at some point and integrate all the modules. The work has been divided into smaller tasks and the members have worked parallel and then after each milestone they have discussed the status of the project and continued with the other subtasks.
This was a pretty democratised team because the discussions were horizontal in hierarchy and the coach was only making sure everything worked, without interfering drastically in the decisions made from the team members.
The controlling schema was a decentralised one (CD).

5. Chosen process model

To deliver the software with success by the end, there was a need to choose a process model. The chosen model is the *XP (Extreme Programming)*. A general schema of this model is shown in Figure 1.
This model being an evolving one, makes the work easier when wanting to make changes even during the different stages of development and documentation. The changes are evident in the design of the data storage, interfaces and all the components of the system in general. The client was close to the developers and therefore the changes were made right away. This is why this was a XP team.
The whole system had a lot of new technologies to which the team members were required to adapt fast.

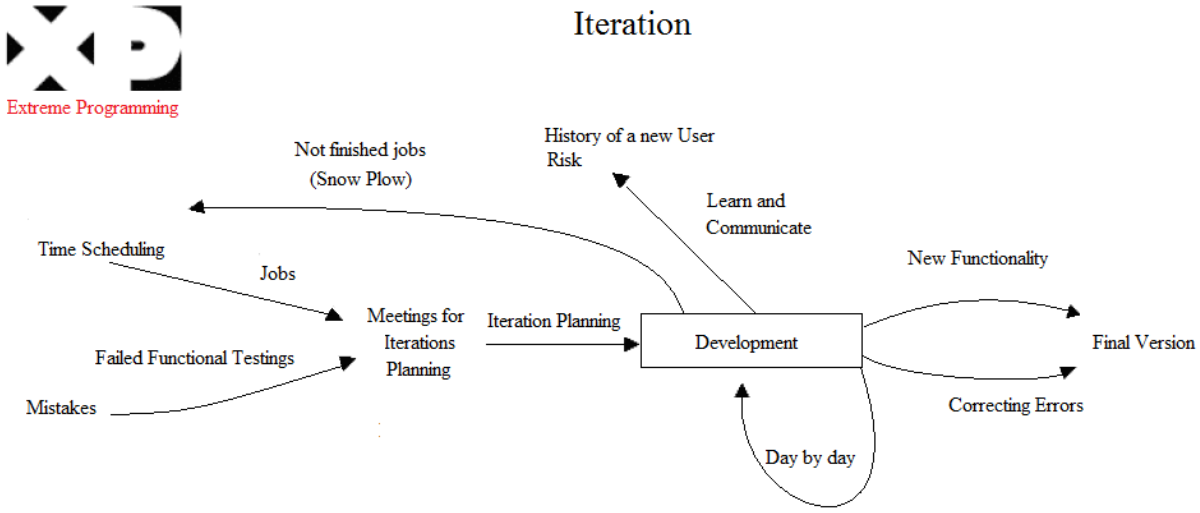


Figure 1: The Extreme Programming process model.

6. Work Flow

This shows the development process from the beginning till the end.
This flow has many iterations and changes during all the stages. It also has some turn back points where there were obstacles and things not working. In the end all the tasks were integrated together. Of course during all the stages there has been continuous testing of the components. The diagram is shown in Figure 2.

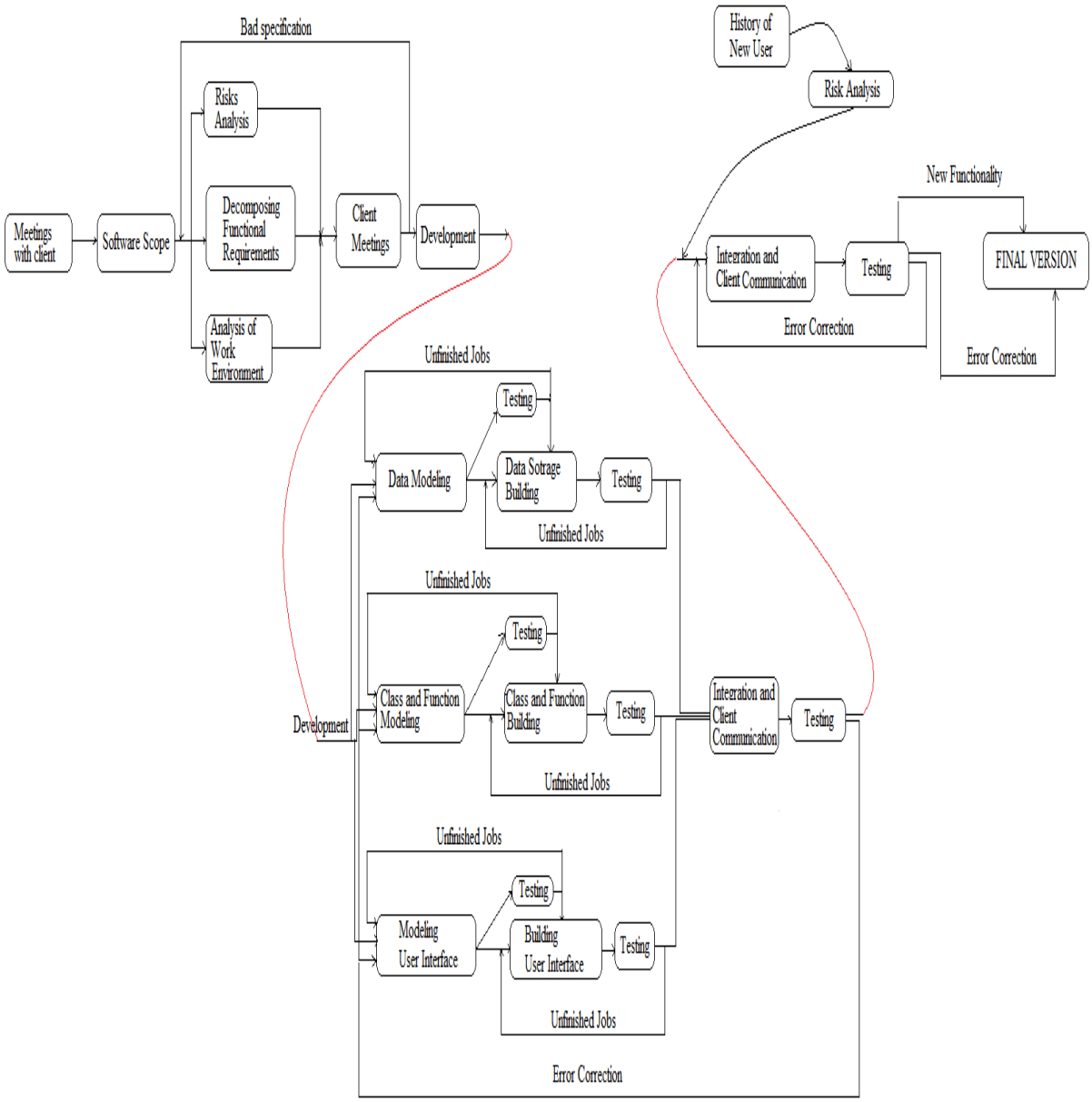


Figure 2: Work Flow

7. Project schedule, milestones and problems

Project schedule:

We had planned weekly meetings to check the state of the project and plan for the next meeting. Each member had assignments to deliver until a precise date.

Milestones:

April/May 2010: Organisation and client problem statement understanding.
June/July 2010: Software engineering modelling.
August/September/October 2010: Coding, testing and documentation.

Problems:

Having chosen the extreme programming as a process model, we have had to loop again and again if we encountered problems during the project progress, what delayed us.

8. Definitions, acronyms and abbreviations

EMF Store: http://www.eclipse.org/proposals/emf-store/
The EMFStore is a model repository that allows to store EMF model instances and which keeps a version history of these instances. EMFStore follows the checkout/update/commit interaction paradigm known from SVN or CVS. The framework allows to checkout a copy of a model instance from the repository. Then it tracks change on the model instances on the clients and provides an API to send the changes to the repository. Also the API allows updating the model instances according to changes of other clients via the repository.

WebDAV: http://en.wikipedia.org/wiki/WebDAV
Web-based Distributed Authoring and Versioning (WebDAV) is a set of methods based on the Hypertext Transfer Protocol (HTTP) that facilitates collaboration between users in editing and managing documents and files stored on World Wide Web servers. WebDAV was defined in RFC 4918 by a working group of the Internet Engineering Task Force (IETF).
The WebDAV protocol makes the Web a readable and writable medium, in line with Tim Berners-Lee's original vision. It provides a framework for users to create, change and move documents on a server (typically a web server or "web share"). The most important features of the WebDAV protocol include:

- locking ("overwrite prevention")
- properties (creation, removal, and querying of information about author, modified date, etc ...);
- name space management (ability to copy and move Web pages within a server's namespace)
- collections (creation, removal, and listing of resources)

VoIP: http://en.wikipedia.org/wiki/VoIP
Voice over Internet Protocol (Voice over IP, VoIP) is a general term for a family of methodologies, communication protocols, and transmission technologies for delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Other terms frequently encountered and often used synonymously with VoIP are IP

telephony, Internet telephony, voice over broadband (VoBB), broadband telephony, and broadband phone.
Internet telephony refers to communications services - voice, facsimile, and/or voice-messaging applications - that are transported via the Internet, rather than the public switched telephone network (PSTN). The steps involved in originating an VoIP telephone call are signaling and media channel setup, digitization of the analog voice signal, optionally compression, packetization, and transmission as Internet Protocol (IP) packets over a packet-switched network. On the receiving side similar steps reproduce the original voice stream. VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. Codec use is varied between different implementations of VoIP (and often a range of codecs are used); some implementations rely on narrowband and compressed speech, while others support high fidelity stereo codecs.

Axis camera: http://www.axis.com/
Based on open IP standards, Axis network cameras connect to any kind of IP network, including the Internet, and enable remote viewing and recording from anywhere in the world. They also provide advanced video analytics features, such as motion detection, audio detection and tampering alarm.

9. References

Client: EADS.
Aeronautic Defence and Space Company N.V. (EADS) is a large pan-European aerospace corporation, formed by the merger on 10 July 2000 of DaimlerChrysler Aerospace AG (DASA) of Germany, Aérospatiale-Matra of France, and Construcciones Aeronáuticas SA (CASA) of Spain. The company develops and markets civil and military aircraft, as well as communications systems, missiles, space rockets, satellites, and related systems. The company is headquartered in Leiden, the Netherlands, and operates under Dutch law.
http://en.wikipedia.org/wiki/EADS

Chair for Applied Software Engineering
http://www1.in.tum.de/static/lehrstuhl/

Prof. Bernd Brügge, Ph. D.
http://www1.in.tum.de/static/lehrstuhl/de/people/professor

Dipl. Inf. Univ. Michael Nagel
http://www1.in.tum.de/static/lehrstuhl/index.php/people/102-michael-nagel

Michael Huber
http://www1.in.tum.de/static/lehrstuhl/index.php/people/215-michael-huber

Lectures:

B.Bruegge, A.H.Dutoit, Object-Oriented Software Engineering Using UML, Patterns, and Java, 3nd Edition, Prentice Hall, Englewood Cliffs, NJ, September, 2009.

Apple iOS Dev Center
http://developer.apple.com/devcenter/ios/index.action

II. Requirements Analysis Document

1. Introduction

1.1 Scope of the system

This project aims to build a software which makes the maintenance work as electronically as possible.
This software will be used from the Line Maintainers and Experts positioned in different places of the world.
The user interface should be friendly and fast to work with since the workload is regarded to be intense.
The software itself offers the possibility to access the different types of aircrafts, their parts, and also add different kinds of attachments to each of these parts.
It also allows voice conference between the Line Maintainers and the Experts and generally speaking; interaction between the users. Technically speaking in general terms, it has different connectors which can load the elements of the aircrafts and connect to the attachments related to these parts which on the other hand are also connected to sessions.

1.2 Objectives and success criteria of the project

Input data

The software at the beginning has a complete list of aircrafts and their parts, a list of sessions and a list of users and attachments connected to both sessions and aircrafts.
The list of users can be independent from the session.
For each aircraft element an ID, a name and a description are given. Each node might have other child nodes. For example the node of a type of aircraft might have child nodes which are parts of the aircraft such as wings, engines, etc ...
In the system are many lists registered like lists of the users and their credentials, lists of the users who want to participate in a certain session, lists of the sessions and their types, etc ...

Output data

After all the data is registered in the system, they are shown in the application as a list from which the user can select each of them and edit them too.

Data management

The flow of the input information until its final computation goes through from storage to displaying it. The first stage is to design and model the data. This shows high risk because all the other processes will depend on this. The software needs a precise and known structure of the data so that it can be used correctly from the programming language used.

Performance

Fast generation of output data, filtering and structuring it as the user demands it.
Fast connection to the data storage servers for updating the data.

Decomposition in Main Functions

Below are shown the main functionalities of the system:

Register the users by categories.
Store the aircraft types and their parts.
Strore the attchments related to the aircraft types and parts.
Register the sessions and their types.
Register the users to the sessions.
Enable Voice conference.
Enable Video Streaming.
Show status of sessions.

2. Current system

The actual system is called Armed and is a desktop application. This means it is not mobile and it cannot be accessed from the actual working place where the reparation is actually performed. The data is not saved directly into the system during the inspection phase and this might lead to some missing information during the After-Flight procedure. It is not possible to take pictures and save them, only the narrative description is possible. The assistance in distance is not possible, and the Expert has to travel until the actual place where the problem occurred to fix it instead of voice assisting the Line Maintainers. All these delays might be critical in some situations.

3. Proposed system

The iArmed system has a certain requirement and vision of how it should work. Below is shown the general overview of the system and some other details.

3.1 Overview

General overview

The system has different main components. First of all there is a need for the iPhone which will be mostly used from the Line Maintainers but maybe also from some Experts. Then is the iPad which will be used only from the experts. The iphones need to connect to the iPad and establish a video conference. In the beginning the problem was whether the iPad could handle many video conferences at the same time. And this has to still be tested, but most likely it will handle only one and there fore the Expert has to switch from one video streaming to the other. All the iPhones have to connect to the EMF Store Data Storage Server to retrieve the aircraft models and also to the WebDAV which has stored the attachments related to the elements of the EMF Store. There are many risks to take care of such as wether the exportation of the models from the iPhone to the WebDAV will be possible in the correct format and reading the data in the other way as well or not. Then the VOIP should be tested as well because chances are that the connection might not be able to be established.

Techniques for gathering data information

Gathering the basic information of what the final product should look like was easy since the client was clear of what he wants. There was need to discuss technical details with the client regularly.

The client also introduced the previous system to better understand how it was working and what is needed to be changed, added or not used anymore.

Assumptions and dependencies

The system requires iPhone with iOS4.1, iPad, Mac platform for development, WebDAV Connector, HTTP Connector, VOIP Connector, Asterisk Server for VOIP conferences, AXIS cameras for video streaming.

Restrictions

The team had never experience working with such a system, so the lack of experience was one of the greatest restrictions. The testing will be done with only a few devices for practical and budget reasons.

3.2 Functional requirements

Configuring the system

Registering the aircrafts and their parts.
Registering the users and their categories (Line Maintainers and Experts) together with their login accounts.
Registering all the devices in the system such as iPhone, iPad, AXIS camera, and testing the connections.

Managing sessions (After-Flight & Inspection)

After-Flight session is due after every aircraft lands.
Create a new session, edit and delete it.
Join or leave a session.
Add or delete an user from a session.
Create and view valid EMF Objects conform to the armed.ecore specification.
Select aircraft based on type and call sign.
Create and view annotations for the EMF Elements such as Video, Picture, Text and save them to WebDAV.
Automatically reload the EMF Model if something changed.
Browse EMF Model and WebDAV attachments.
Show the users connected to a session.

Managing repair sessions

Schedule session.
Initiate a VoIP conference call after joining a session.
User sets his status to ready.
Expert watches a live video stream from an AXIS camera.
Users add new camera to the system.
Expert switches between different video streams.

3.3 Nonfunctional requirements

3.3.1 Usability

This system aims to develop an application with user-friendly interface which is practical and easy to be used from the end users. The graphical interface should be intuitive and have as fewer steps as possible for a task by asking the user only the necessary input and giving him the right feedback. Also the environment of data entry should be familiar to the user, meaning that it should resemble the way he is already working.

3.3.2 Reliability

The application should not create any XML files on the server; instead EMF objects should be created.
Only authorized users should be able to access the data.

3.3.3 Performance

The system should always be responsive to user inputs or show a progress indicator.
It needs to be usable while working at or inside an aircraft.
There should not be noticeable delays when polling the data from EMF Store and WebDAV, no big delays in video streaming and VOIP conferences.
Also saving all the modifications done in a session should not wait for a long time for a feedback. The system should notify if there is a failure, the operation succeeded or it is still waiting for a response.

3.3.4 Supportability

This product needs the support from the client represented by the chair of applied software engineering. The requirements are clear but there is a need for maintenance from the client side too. It needs also a lot of technical assistance because this is not a typical commercial application. Problems might arise in the future.

3.3.5 Implementation

It must run on Apple´s iPhone iOS 4.1. Therefore Xcode Cocoa programming is announced.

3.3.6 Interface

The iPhone interface is quite easy and intuitive to develop for but updating Apple´s SDK may arise the need of some changes in the graphical interface and functions.

3.3.7 Packaging

The application should be complete, precise, unambiguous, consistent in the implementation, independent and verifiable. It is important that it is modifiable, readable, and ready for reference or review in the future.

3.3.8 Legal

Legally speaking, this software should not have problems. Perhaps an agreement regarding the importance of using the software from the users is needed. But this is not our concern at the moment.

3.4 System models

3.4.1 Scenarios

After-Flight Inspection – Line Maintainer

After the aircraft has returned from a flight the Line Maintainer has to perform an After Flight Inspection.
He starts the iArmed application and starts a new Session. He enters a session name and selects the Voice annotation feature and gets an overview over the model of the aircraft. The maintainer starts his inspection and finds a damage at the left wing. He puts on his BT headset, takes his iPhone and selects the "Wing, Left" Element from the aircraft model. Then he starts recording and gives a short description of the damage. Then he stops the recording.
The maintainer decides that he has to further investigate the damage, therefore he has to enter a maintenance manhole. As his movement is very restricted and the maintainer won't be able to operate the iPhone during the inspection in the manhole, he starts the recording on spec. He finds a damage at the fuel pipeline and describes it. After he has left the manhole the Line Maintainer saves the recording to the "Wing, Left - Fuel Pipeline" Model Element.
The Line Maintainer finishes the inspection without any further findings so he closes the session and closes the app.

Structural Damage Inspection Session – Line Maintainer

The Line Maintainer starts the iArmed app. He gets a list of planned repair sessions that he is supposed to participate. He chooses the "Wing Damage Inspection" Session that is supposed to start in 30 Minutes.
The Line Maintainer gets a list of participants. He can also see a list of attachments like pictures, repair instructions or videos. The Line Maintainer opens the document "Wing Damage Repair Preparations.doc" and follows the contained instructions.
At the time the session starts the other participants join the session. When the Maintainer is ready to begin the repair, he sets his status to ready.
The application automatically connects to an asterisk conference call with all participants via VoIP.
The maintainer is instructed to take a picture of the damage and share it with the other participants. He takes his iPhone and uses the built in camera to take a picture. After taking the picture the maintainer gets a preview so he is able to check if the picture is showing the damage properly. The picture isn't sharp enough so he chooses to take a new picture. The new picture is perfect so he decides to use it.
As the damage is hard to see the maintainer wants to annotate the image to point out to the other participants where the damage is located. He chooses the annotation mode and draws a circle around the damage. Then he adds a text note to the picture. After that the image is ready for sharing so the maintainer selects "Share Image".
The software proposes a name for the picture "20100421-1603 - Wing Damage Inspection - Picture 1". The Maintainer could change the name but decides that he wants to use the proposed one and confirms. The Image file is sent as attachment and linked to the current session. Other participants can then find and open the image file in the file attachments list.

The maintainer continues with the examination of the damage and gathers additional information. He wants to link this information to the image file that he has attached to the session before. He selects the image file and chooses the "Add voice annotation" functionality. The iArmed app shows the voice annotation UI and the maintainer starts recording. He is talking with his bluetooth headset activated and describing the additional information he found. After he has finished he selects stop and can listen to the audio file again. He decides that the recording quality is good and sends the audio annotation to the server, which attaches it to the image file.
The maintainer has gathered all necessary information about the damage, so the inspection session is finished.
He selects leave session and closes the iArmed app.

Structural Damage Repair Session

A remote maintenance session is scheduled to start in 5 minutes. Due to high traffic the expert is not able to reach his workplace in time. He decides to leave the autobahn and park his car in order to join the session with his iArmed app. He opens the iArmed app on his iPhone and selects the "Structural Damage Repair" session. After joining the session he sets his status to ready and is automatically joining the asterisk conference call via VoIP.
The maintainer has provided a live video source using an AXIS Video server that can stream video via MotionJPEG or H264. In order to be able to supervise the repair of the maintainer he chooses to open the video stream and can watch it on his iPhone.
The expert gives the maintainer instructions via the conference call until the repair is finished. After that he leaves the session and closes the iArmed app.

3.4.2 Use case model

Startup application and login

Use case name	**Startup Application and login**
Initiating Actor	**User {LineMaintainer or Expert}**
Participating Actors	User
Entry Conditions	System running; internet connection; ...
Flow of Events	1. User clicks on iArmed Application 2. System displays Login Form 3. User fills login form 4. System checks Login Data & displays Interface

Perform an After-Flight inspection

Use case name	**Perform an After-Flight Inspection**
Initiating Actor	**{LineMaintainer}**

Participating Actors	{LineMaintainer}
Entry Conditions	System running; internet connection; User logged in; ...
Flow of Events	1. LM selects “After-Flight Inspection” 2. System displays a list of AirCrafts 3. LM selects an AirCraft 4. System displays the selected EMFModel 5. LM browses the EMFModel & selects an Element 6. System displays the selected Element → include Attach Media
Exit Conditions	Media saved to the WebDAV Server & URL saved to the EMFStore Server

Create session

Use case name	**Create Session**
Initiating Actor	**User {LineMaintainer or Expert}**
Participating Actors	User
Entry Conditions	System running; internet connection; User logged in; ...
Flow of Events	1. User clicks on Create Session 2. System displays a Form {Type of Session; Users participating; Location; Date; Start Time; End Time; EMFModel; Comments} 3. User fills the form out & commits 4. System uploads the Session to the Server

Join inspection session

Use case name	**Join Inspection Session**
Initiating Actor	**{LineMaintainer}**

Participating Actors	{LineMaintainer}
Entry Conditions	System running; internet connection; User logged in; Inspection Session been created; ...
Flow of Events	1. LM clicks on “Join Inspection Session” 2. System displays a list of Inspection Sessions 3. LM selects an Inspection Session 4. System displays the EMFModel linked to the selected Session & a list of Participants 5. LM sets his Status to READY 6. System updates the user status & builds a Conference Call with READY participants → include Browse Media → include Attach Media

Join repair session

Use case name	**Join Repair Session**
Initiating Actor	**User {LineMaintainer and Expert}**
Participating Actors	User
Entry Conditions	System running; internet connection; User logged in; Repair Session been created; ...
Flow of Events	1. User clicks on “Join Repair Session” 2. System displays a list of Repair Sessions 3. User selects a Repair Session 4. System displays the EMFModel linked to the selected Session & a list of Participants 5. LM sets his Status to READY 6. System updates the user status & builds a Conference

		Call with READY participants → include Browse Media → include Attach Media
	7. LM turns on the AXIS Camera	
		8. System displays the Video on the iPhone/iPad

Review annotations

Use case name	**Review Annotations**	
Initiating Actor	**{Expert}**	
Participating Actors	{Expert}	
Entry Conditions	System running; internet connection; User logged in; ...	
Flow of Events	1. E clicks on Review Annotations	
		2. System displays a list of EMFModels
	3. E selects an EMFModel	
		4. System downloads the EMFModel and displays it → include Browse Media → include Attach Media

Browse media

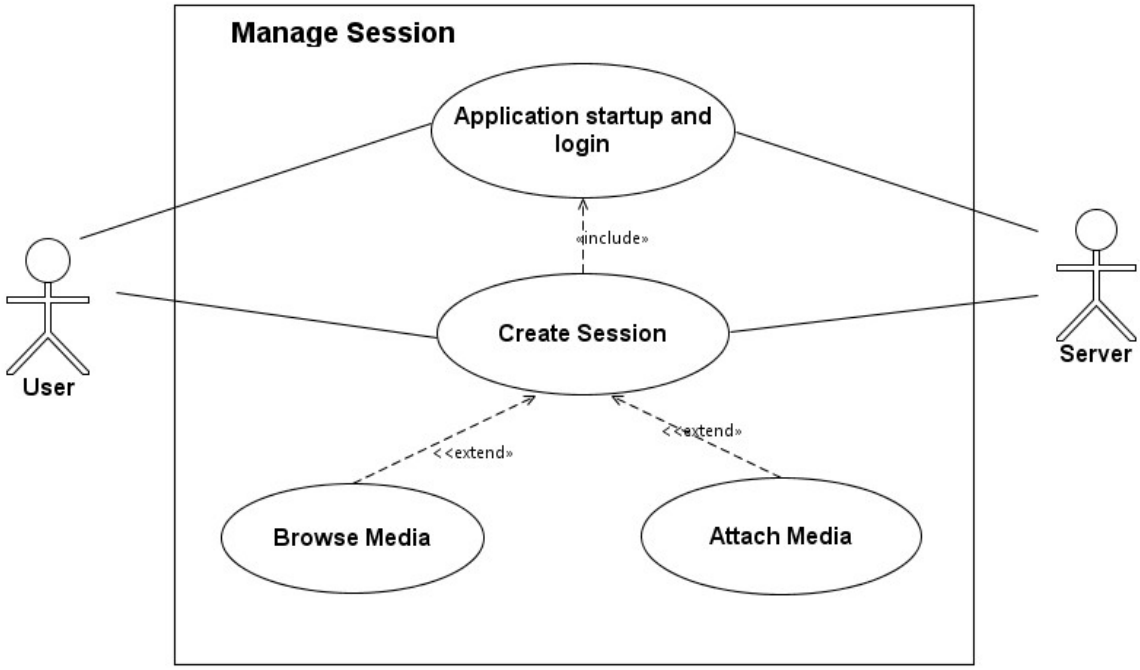
Use case name	**Browse Media**	
Initiating Actor	**User {LineMaintainer or Expert}**	
Participating Actors	User	
Entry Conditions	System running; internet connection; User logged in; ModelElement selected; ...	
Flow of Events	1. User clicks on “Browse Media”	
		2. System loads & displays a list of Media from the WebDAV Server
	3. User selects an Attachment	
		4. System displays or plays it

Attach media

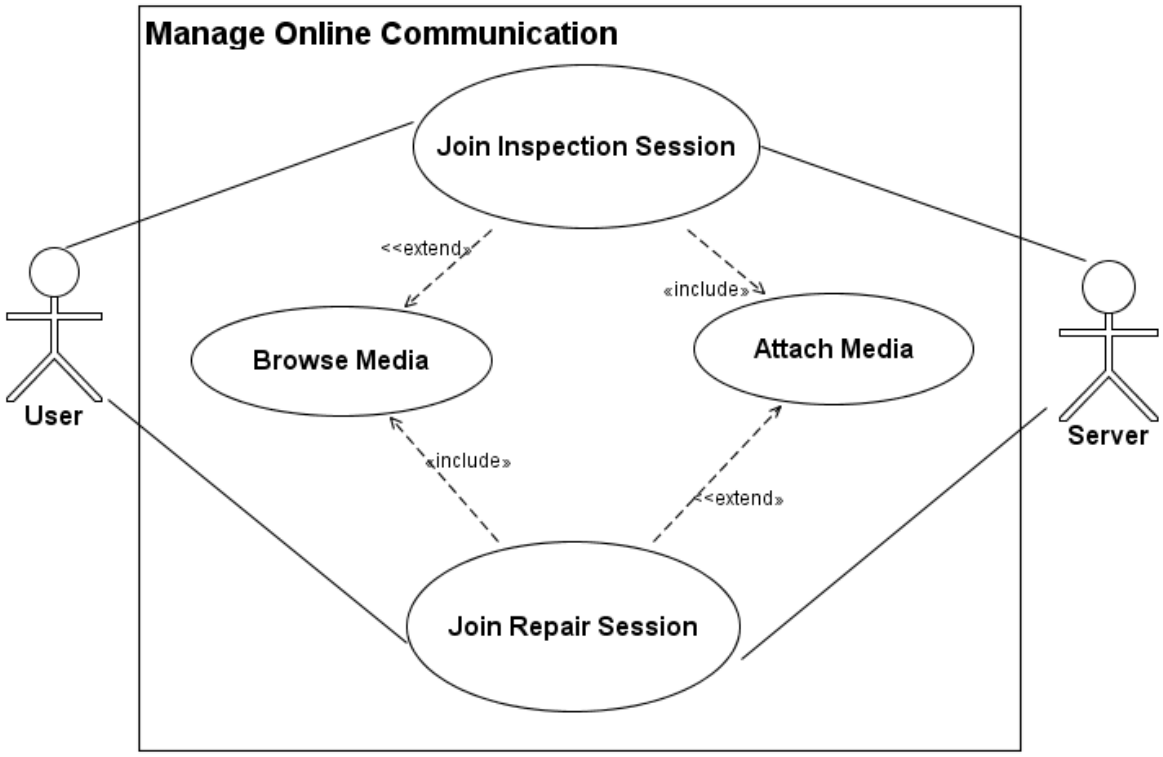
Use case name	**Attach Media**
Initiating Actor	**User {LineMaintainer or Expert}**
Participating Actors	User
Entry Conditions	System running; internet connection; User logged in; ModelElement selected; ...
Flow of Events	1. User clicks on “Attach Media” & selects type of Media {Text; Picture; Video; Voice Annotation; …} & selects Media to upload 2. System sends the file to WebDAV Server & attaches URL to EMFModelElement on EMFStore Server

Use case diagrams

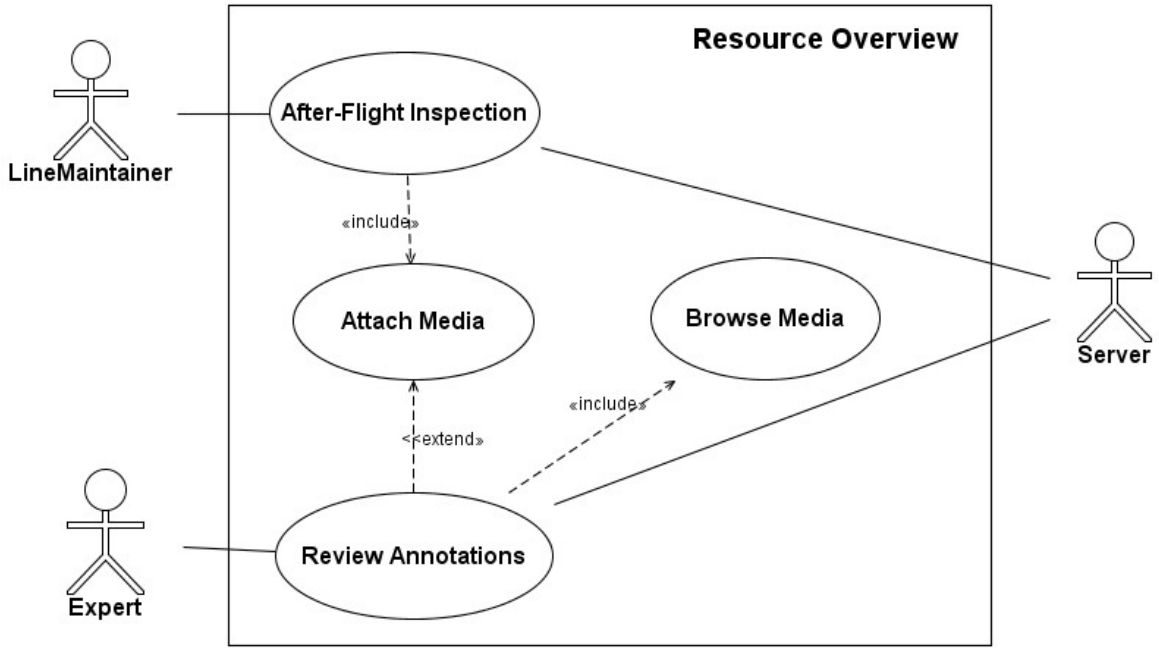
Manage session



Manage online communication



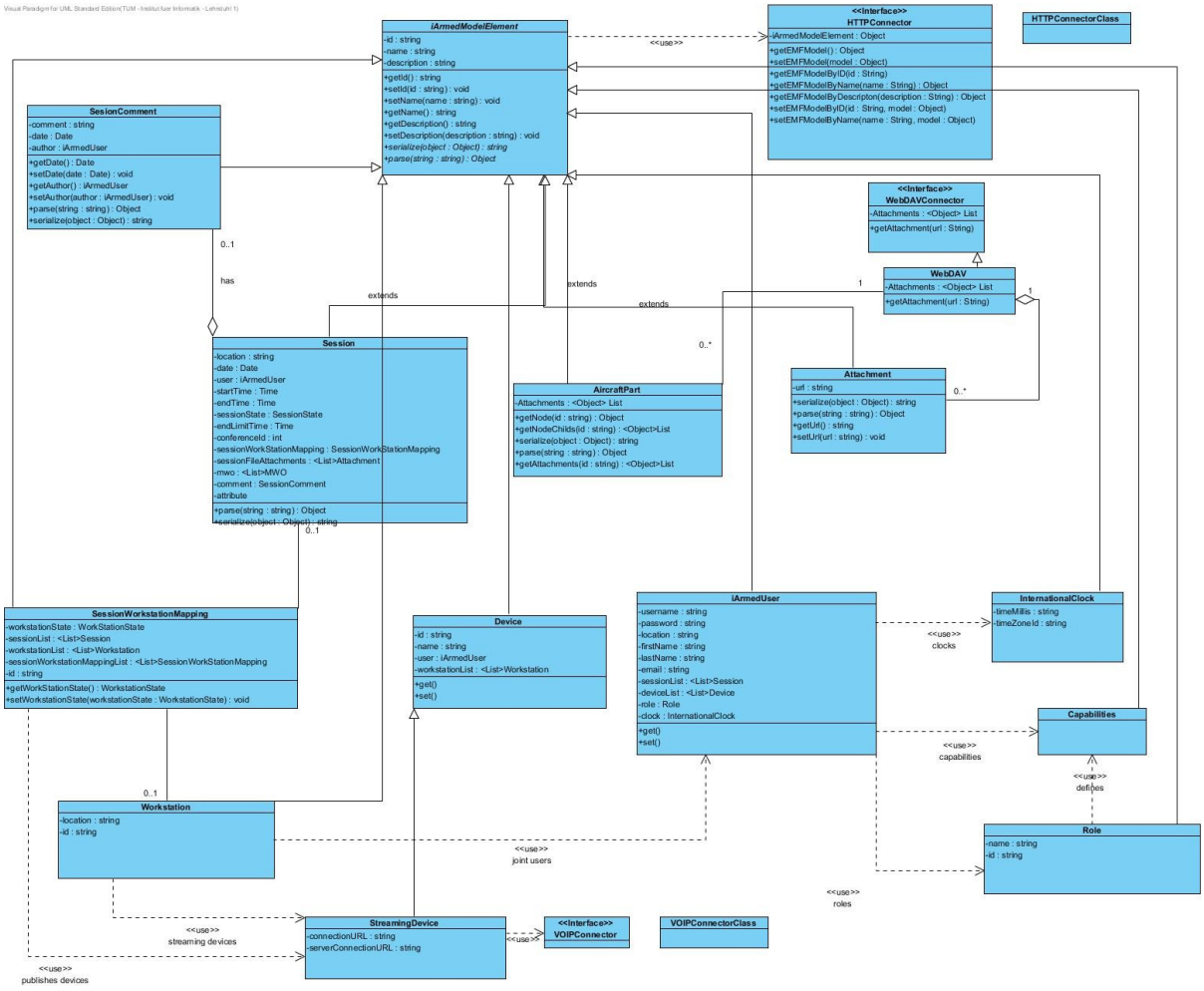
Resource overview



3.4.3 Object model

Class diagram

should be updated!

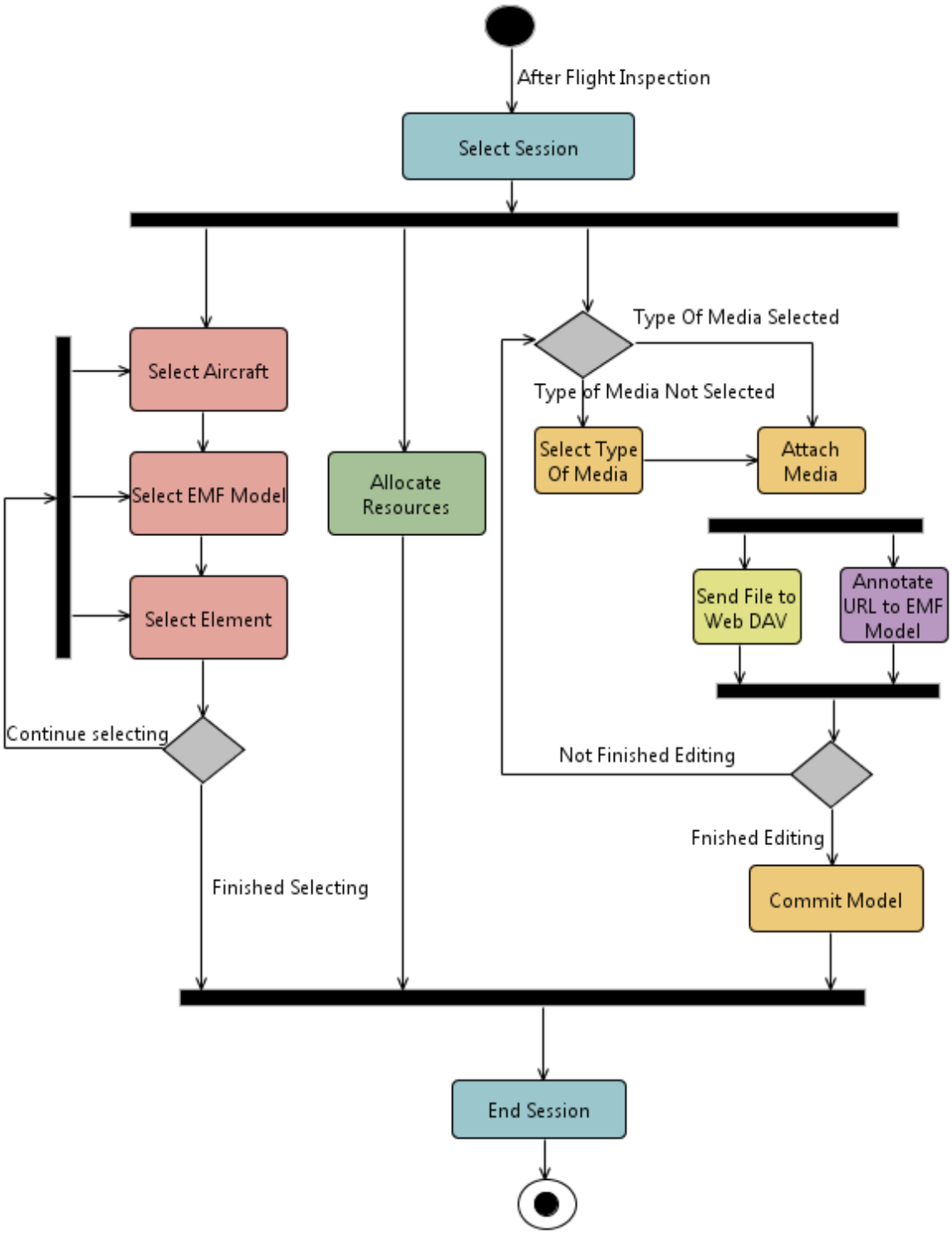


Object diagram

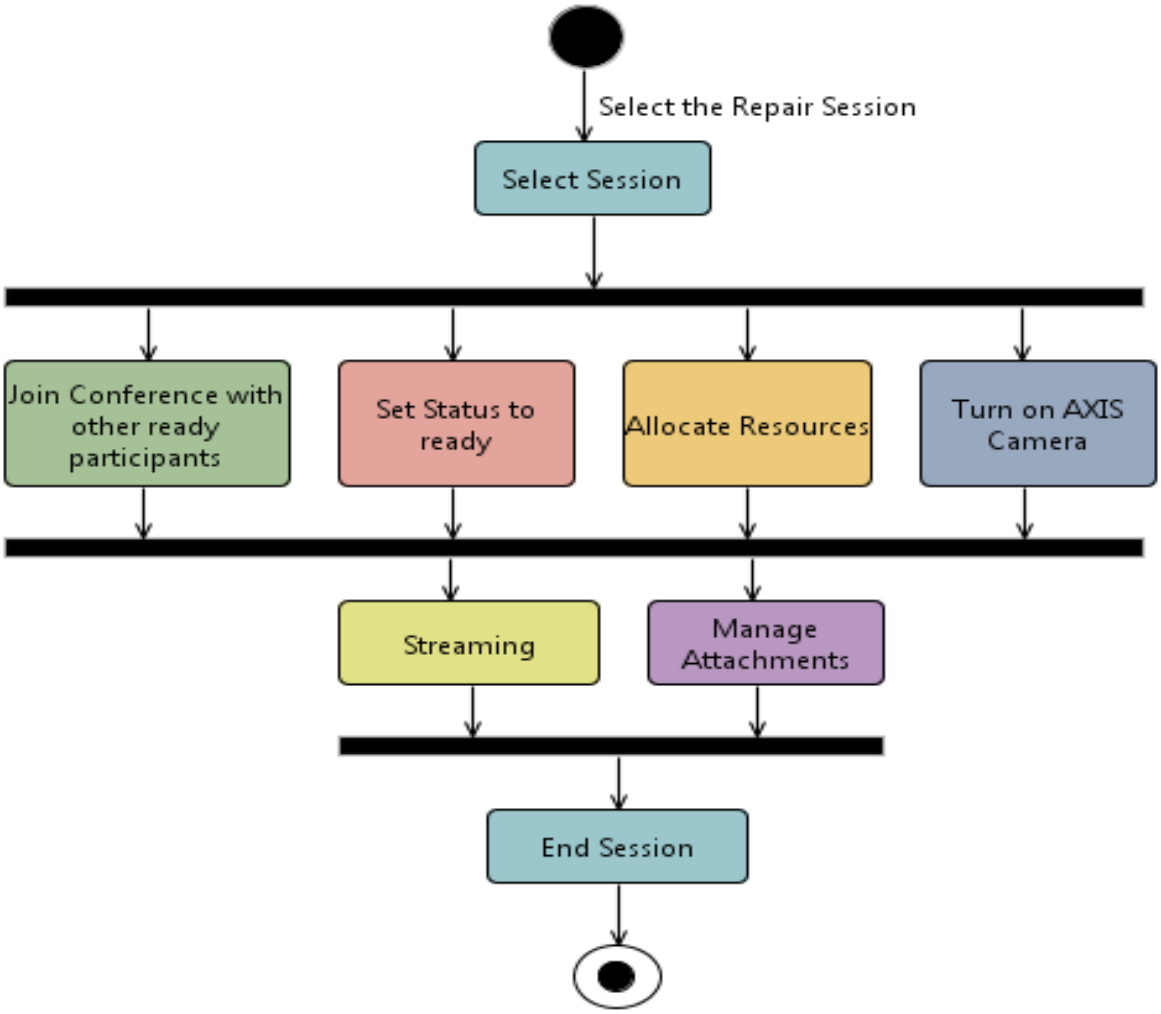
coming soon, being updated from Benedikt!

3.4.4 Dynamic model

After-Flight inspection activity diagram

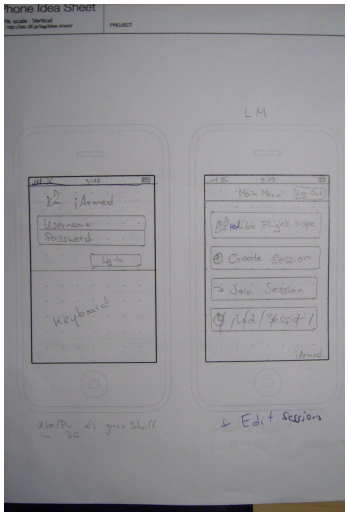


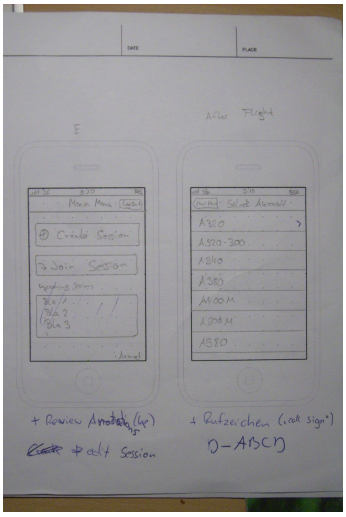
Join repair session activity diagram

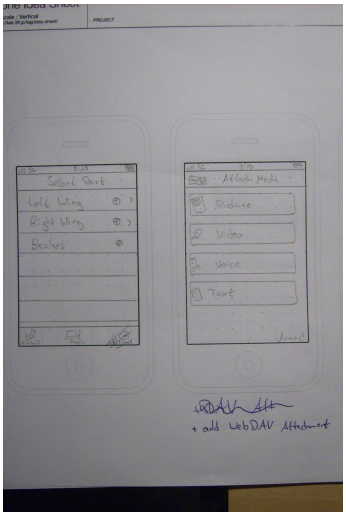


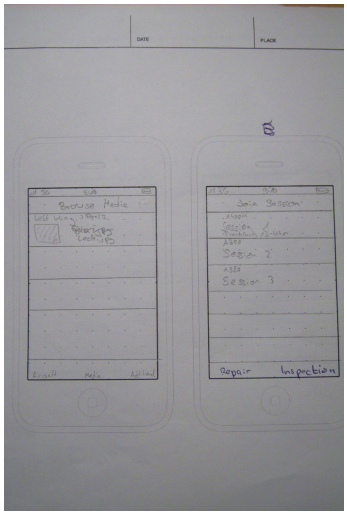
3.4.5 User interface, navigational paths and screen mock-ups

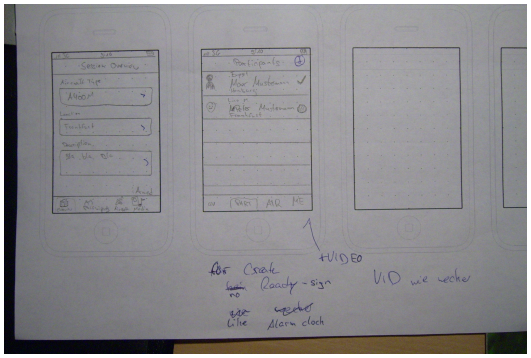
Paper prototypes





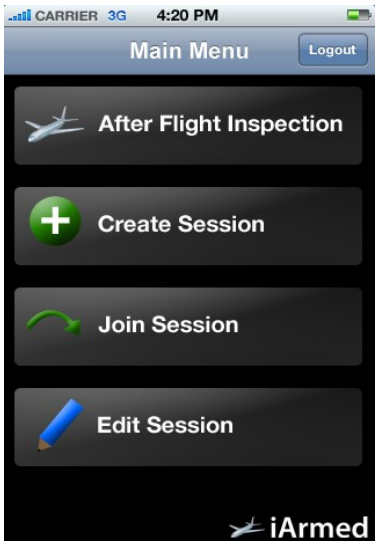


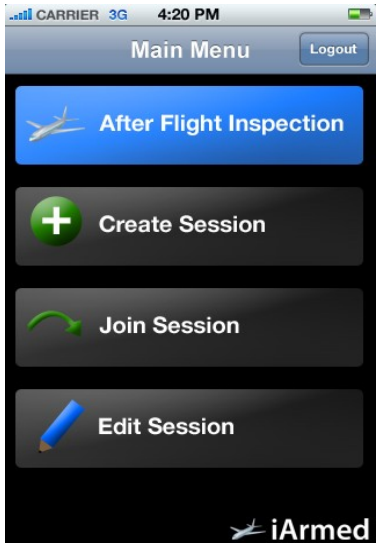


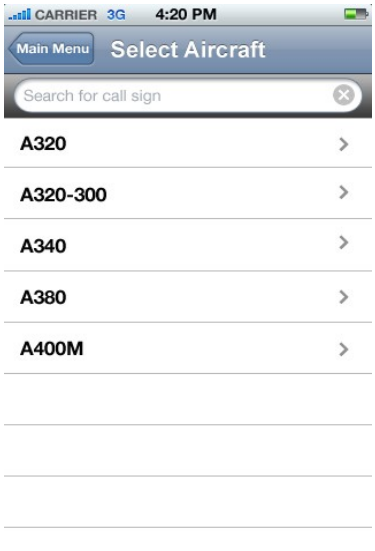


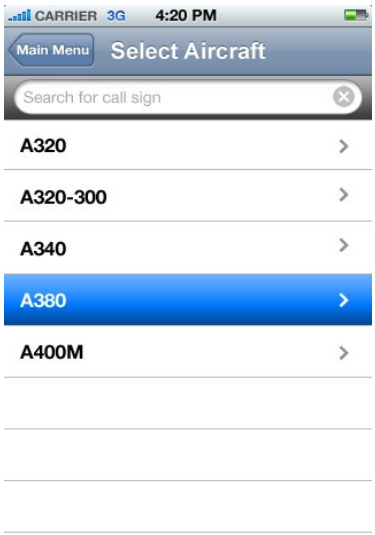
Screen mock-ups

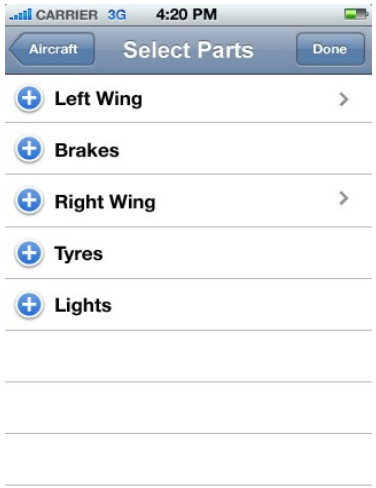
After-Flight inspection scenario

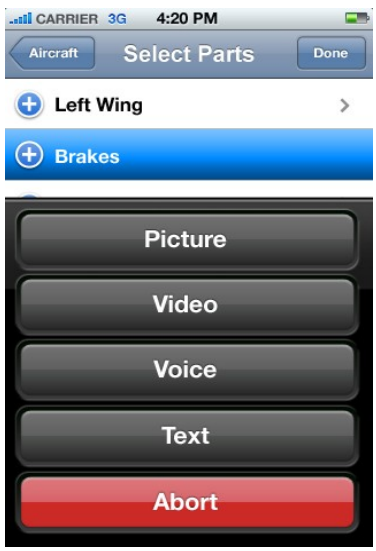




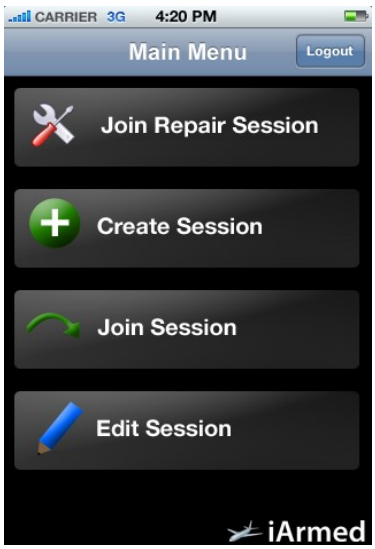


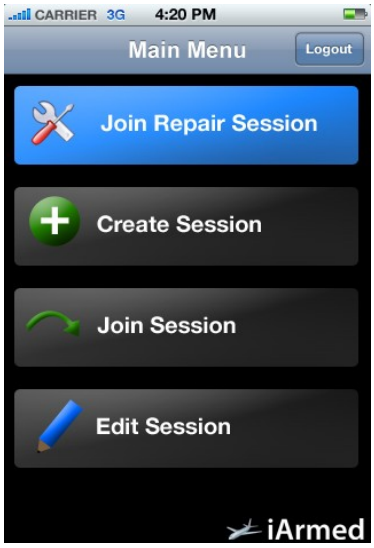


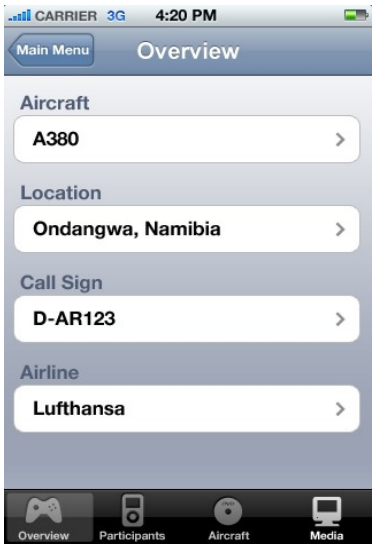




Repair session scenario



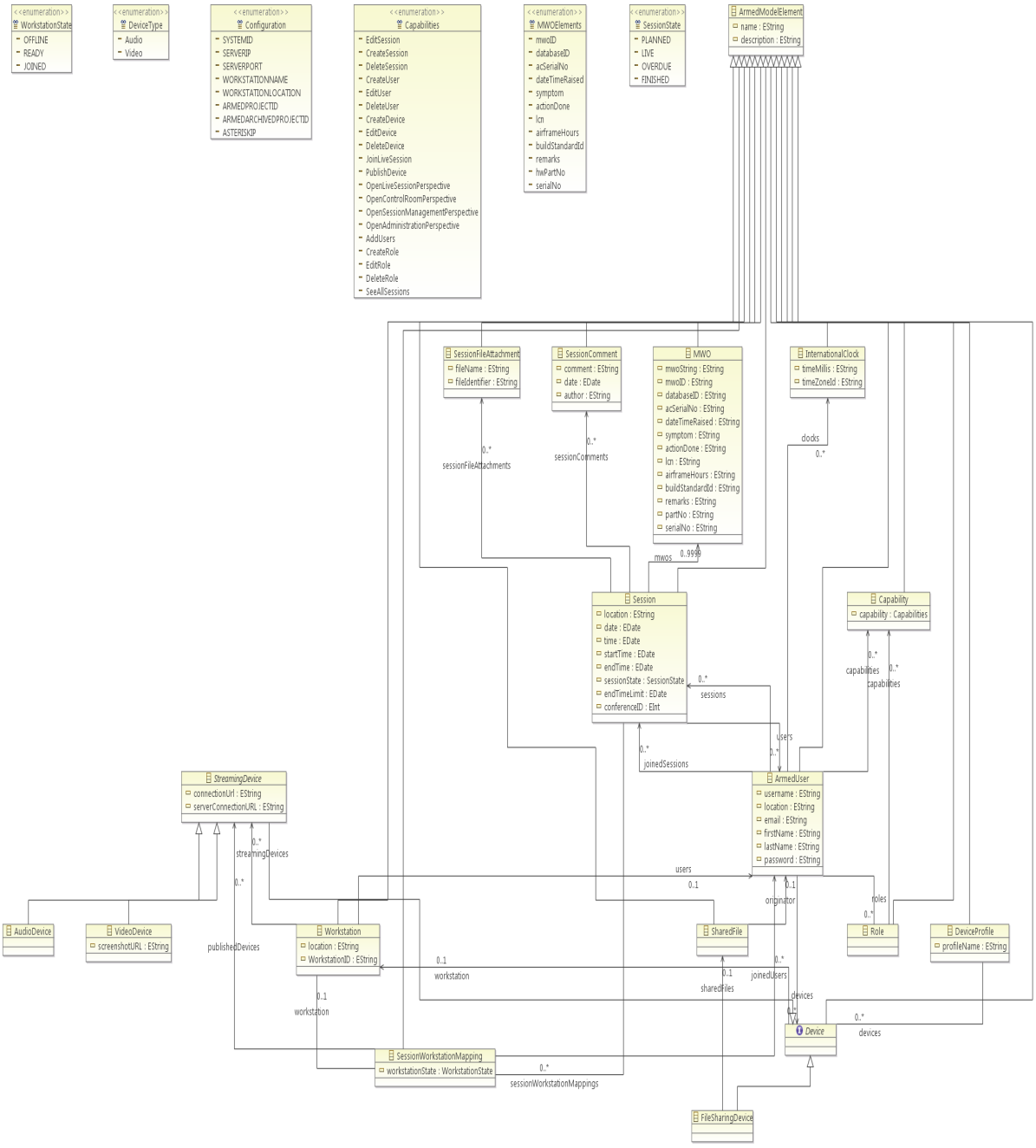




III. System Design Document

1. Current software architecture

The current system is a desktop based system.



2. Proposed software architecture

Architectural modelling

The model is the one named „Architecture with five levels“. The elements are as follow:
Presentation
The information which comes from the servers is presented in the client application.
User interface
There are different user interface views for the Experts and the Line Maintainers. The experts may have also the interface on the iPad which the LM do not have at all.
Business logic
The users of the system are mainly technician who know what to do. The only thing missing is accustoming to the use of iPhones and iPads. Basically we are dealing with trained users. The businesses focus in on making the application work and have good performance. The visual details are secondary.
Managing and control
The communication should be synchronised so all the components of the system should communicate all together at same point. The EMF models can be updated every few seconds on the client presentation, but we should make sure that all the updates are shown on time to each client. All the users should be notified at the same time if some change occurs. The video and voice streaming might have some delays, but the users should be notified when the connection is established. The other users should be notified when a new participants enters or leaves the conference room.
Data storage
The storage is done in two different servers. One saves the EMF models and the other the attachments related to these models. The logic is different. In the EMF Store there is a tree model, while in the WebDAV server which contains the attachments there is only a link and the link is referencing a model, so there is no specific structure.

2.1 Overview

SCD diagram

The Figure 6 Shows the SCD diagram which shows the general overview of all the tasks of the system and the way the user interacts with the system. The system is composed of five main parts which are:

Input data: The input information is the user authentication and the EMF elements information. This part communicates with the Data storage system through ID, name, description, username and password. According to this input, the system generates the proper output. This output is then transferred to the graphical system.

Communication Interfaces: There are two types of interfaces, the iPhone and the iPad. The information can be shown on both interfaces. The difference is that on iPad the user can see the video streaming which is sent from the Axis Camera.

Output Data: This part generates the output information according to the input and user requests. It can be the status of the user connected to a session, the information of a specific element of the EMF model, the information of a specific attachment.

Maintenance and self testing system: In this module continuous requests and security controls are made. It makes sure the connection is established and there is no erroneous data. Also no interference from third parties should be allowed.

The main part of the diagram makes the information processing together with the device management system. This is the most important part of the system because this realises the logic of the business.

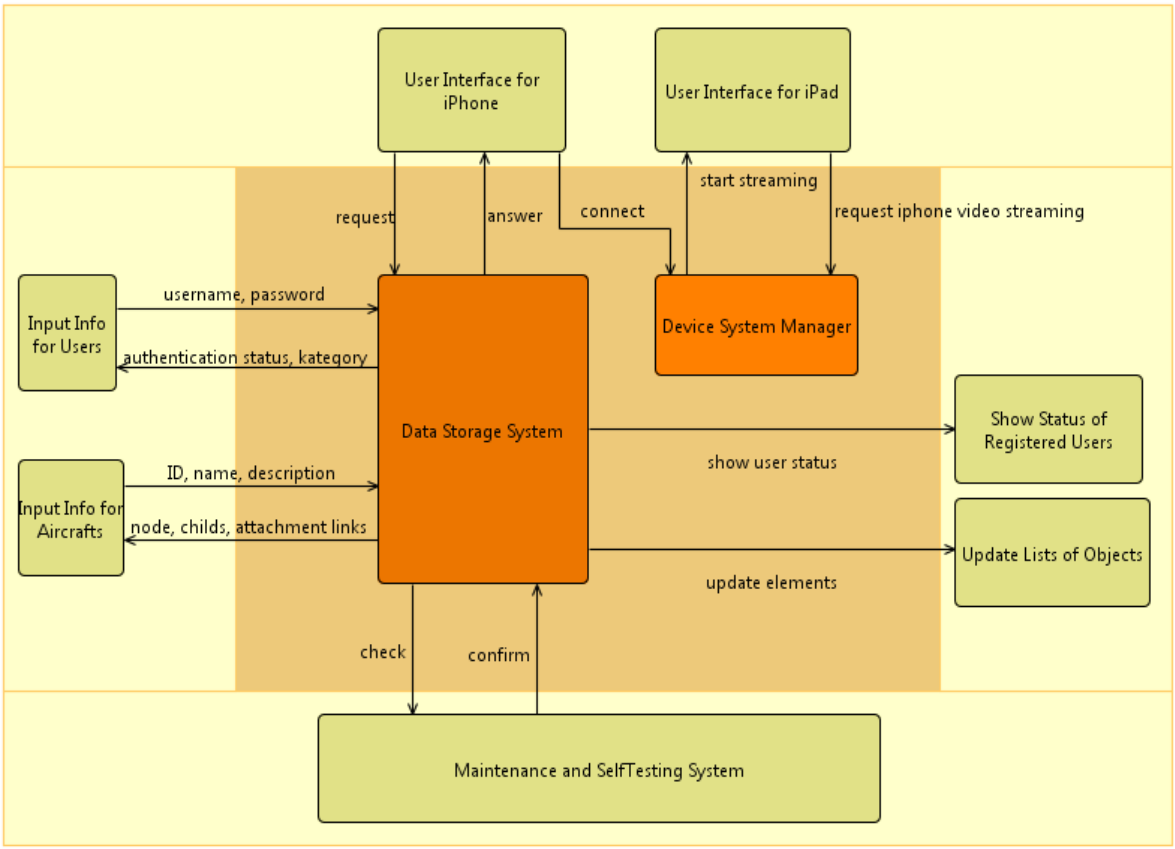


Figure 6: SCD diagram

2.2 System decomposition

SFD diagram

In Figure 7 is shown the SFD diagram which shows the job flow in the system. As a beginning the User logs in the system with a username and a password and this will be checked in the EMF Store. The connection to the EMF store is done through the HTTP Proxy Server. After being identified, the user can make other requests such as browsing through the aircrafts types, adding attachments, joining a session, joining a voice conference, sending video, streaming, etc. Every modification is sent to the output and displayed either on the iPhones or iPads. We do not use the iPad anymore because we tested that it is possible to use streaming of images in the iPhone too. They can be shown through Mpeg files in a web view. All the time the information is checked in the testing subsystem, whether the connection is maintained, the format of the objects is the right one.

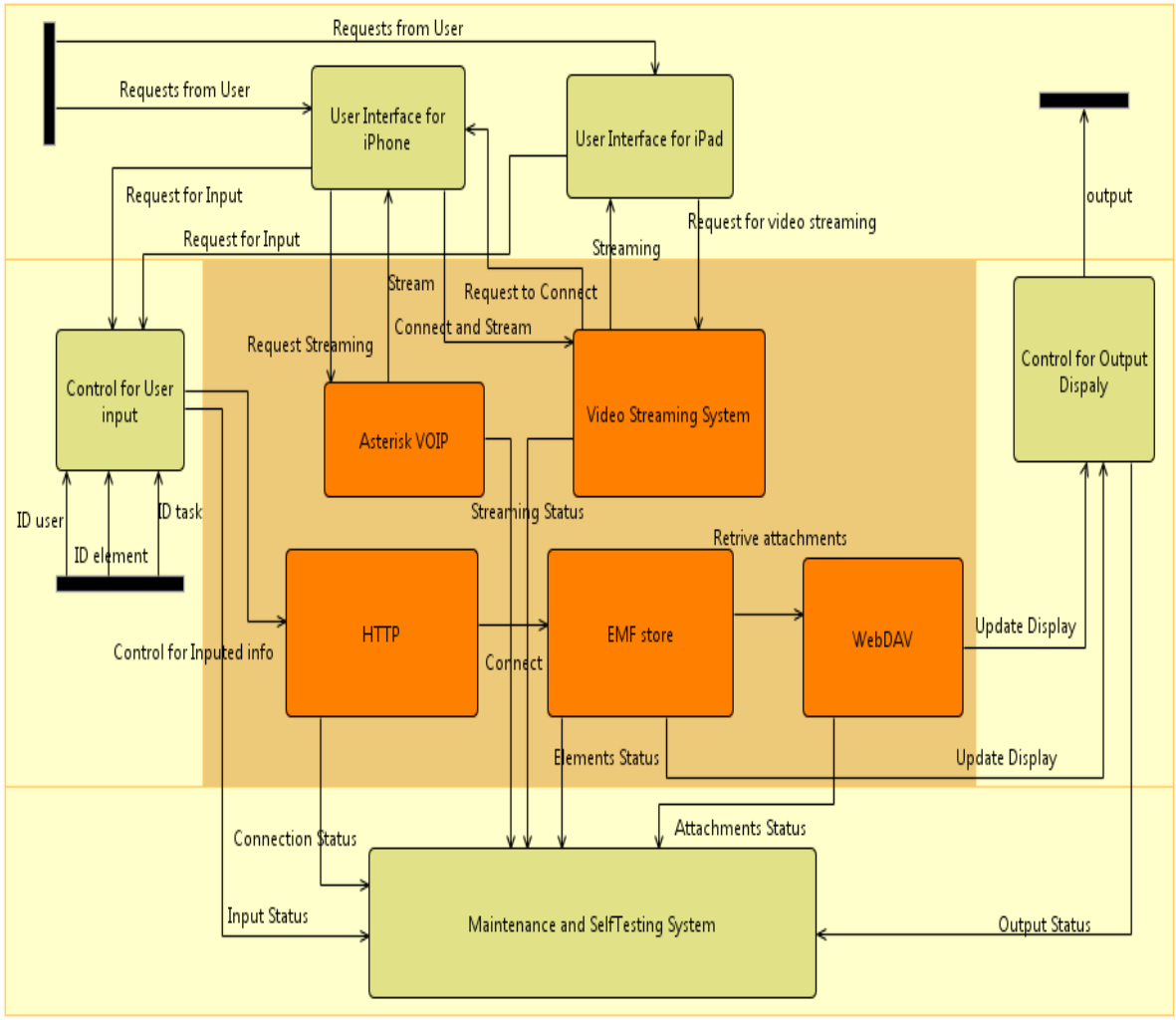


Figure 7: SFD diagram

2.3 Hardware / software mapping

This application is a complex one which uses a certain platform, programming language and hardware. We are using the framework called cocoa-touch. On this framework we are using Xcode as an IDE and the programming language used is Objective-C. The device form the client side is an iPhone. We are using also the camera of the iPhone for photos and videos. The headset of the iPhone are used for VoIP. The rest of the components are the EMF Store which is a data model stored in another server and the WebDAV which is a HTTP server. The address of the WebDAV is http://dav.mrnobody.de/. As for the VoIP server, we are using an Asterisk server which uses a SIP protocol and open source libraries.

Here are some main methods used in each component:

1. VoIP methods:
   {init} – sets up the connection to the VoIP server
   {destroy} – closes the connection to the VoIP server
   {call} – initiates a conference call
   {hangup} – closes the conference call

2. WebDAV methods:
{dav uploadFileFromLocalURL:@"upload.png" remoteURL:remote} – uploads an image, video or text from the iPhone to the remote address of the WebDAV server.

3. EMF store - mainly retrieves the structure of the aircraft, sessions and all the parts related. It shows this on the iPhone app and also saves the addresses of the attachments which are links to the real attachments saved in the WebDAV server.

HTTP EMF Store API

getNode
EMF Object with ID!?
Parameter: ID = EMF Store ID
Example: GET getNode?ID=12345
Output: XML

```
<session>
<location>Munich</location>
        <date>07-03-2010</date>
        <time>04:06:23 UTC</time>
        <mwos>1687</mwos>
        <users>
                <item>123</item>
                <item>456</item>
        </users>
</session>
```

setNode
Save XML as an EMF Object
Parameter: ID = EMF Store ID (optional)

```
<session>
<location>Munich</location>
        <date>07-03-2010</date>
        <time>04:06:23 UTC</time>
        <mwos>1687</mwos>
        <users>
                <item>123</item>
                <item>456</item>
        </users>
</session>
```

Output: HTTP 200 (OK)

deleteNode
Delete an EMF Object
Parameter: ID = EMF Store ID
Example: GET deleteNode?ID=12345
Output: HTTP 200 (OK)

getUserList
Display User list

Example: GET getUserList
<userlist>
<item>1234</item>
<item>2345</item>
<item>3456</item>
</userlist>

getSessionList
Display Sessions list
Example: GET getSessionList
<sessionlist>
<item>1234</item>
<item>2345</item>
<item>3456</item>
</sessionlist>

getAircraftList
Display Aircrafts list
Example: GET getAircraftList
<aircraftlist>
<item>1234</item>
<item>2345</item>
<item>3456</item>
</aircraftlist>

getUser
Get User with Name
Example: GET getUser?username=testuser
<user>
<ID>1234</ID>
<firstname>Test</firstname>
<lastname>3456</lastname>
<username>testuser</username>
</user>

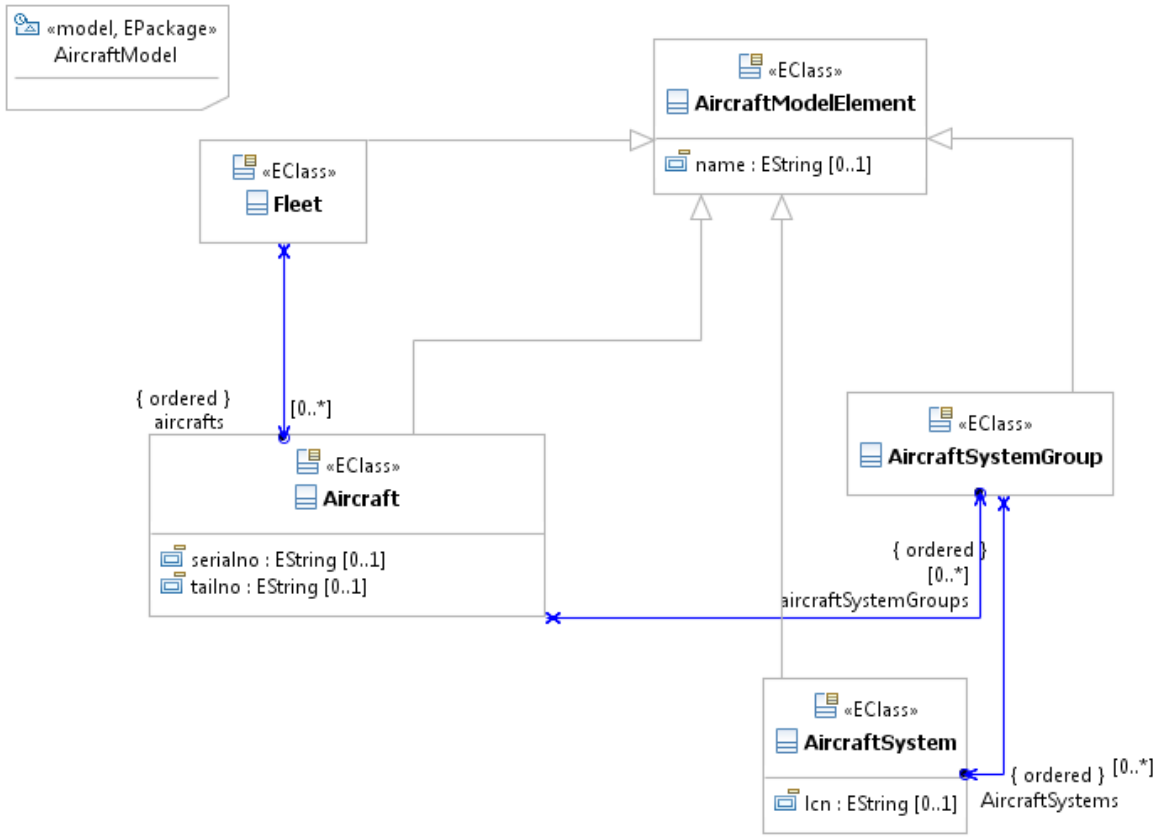
getWorkstation
Get Worksatation with ID
Example: GET getWorkstation?workstationID=
fb891bdb69e52df46ac10079aa31f8aedef7b9ac
<workstation>
<ID>1234</ID>
<workstationid> fb891bdb69e52df46ac10079aa31f8aedef7b9ac</workstationid>
<location>Paris</location>
</workstation>

getModelAttachmentsForSession
Get Attachments for a Session with ID
Example: GET getModelAttachmentsForSession?ID_fb891bdb69_e52df46ac1
<armedfileattachments>
<item>_fb891bdb69_e52df46ac1</item>

<item>_fb891bdb69_e52df46ac1</item>
</armedfileattachments>

Error Handling
401 – Unauthorized: Wrong Login/Password
403 – Forbidden: not allowed action
404 – Not Found: EMF Object not found
500 – Internal Server Error: Server Error

2.4 Persistent data management



2.5 Access control and security

This should mainly be about the way users log in, and how their connection and their profile is loaded. When they log in, if there is some restriction such as the Line Maintainers cannot see everything or do everything but the Experts can. Where do they connect to for accessing data, what they need to access the data.

Everybody sees everything for now. We don't have a search function that is why we cannot filter that easily. The users need a username and a password to login in the EMF Store They receive data such as aircraft lists, the sessions info, the links to the attachments, etc ...

2.6 Global software control

Software Testing and validation

During the testing of the product, we have concentrated in realising the following main points:
The final product is the one the client wants
The final product full-fills all the functions the client requested
The final product is reliable and safe
The final product functions as a whole in a correct way

Testing Techniques

For testing the software we have used Black-Box and White-Box techniques.

Test Cases

Assuring all the possible cases inside each module are tested at least once.
Proof of all the logical predicates either when the result is TRUE or FALSE.
Executing all the cycles with the extreme values as well as the ones within the extremities and checking the results.
We have used the inner structures of the data storage to assure their validity.
Mainly the system has been tested from the graphical interface. During implementation of the modules, we have also tested them by setting input values within the allowed limits and also outside the limits to check the behaviour. We have also checked the performance of each module in any case and have changed parts of the code where errors have been identified. Besides this kind of testing, we have done a general testing to check how well the modules are integrated together. We have also tested the system from another aspect, that in case of malfunctioning of hardware or security threads. For the main functions we have followed all the path from the beginning till the end. We have fully checked the main scenarios. The followed technique is that of debugging.

Examples

Black-Box Technique
Scenario one

-Login in the application either with Line Maintainer or Expert Credentials. (for facility we connect with a guest user with username “super” and password “super”)
-Join a session
-Select a session from the list
-edit the fields
-Join the session
...

Scenario two

-Create Session
-After-Flight Inspection
-Select Aircraft – one from the list

-fill in the data fields
-edit the overview
-add attachments
Results so far: voice messages and pictures can be recorded and saved.

White-Box technique

We add data and we are supposed to read our input back. There is problem in saving the data so far, so we cannot verify the changes we made.

Documentation Testing

This is a very important test for debugging. We will add as many information as possible during the whole development process and afterwards too.

2.7 Boundary conditions

There are many loading delays because we download the data live at the moment. We don't cache information because the peace of data changing is very often. So we need everything in real time from the server. We don't know yet if the system can support large amounts of data, but we hope it supports it. We don't know how many users can connect at the same time, but we don't think there is a very restrict limitation anyways.

3. Subsystem Services

The VOIP device which offers the service through the VOIP Connector.
The saving of attachments which is offered through the webDAV Connector.
The loading of EMF Objects through the HTTP Connector.
For the VoIP we are using an Asterisk Server which uses the SIP protocol.

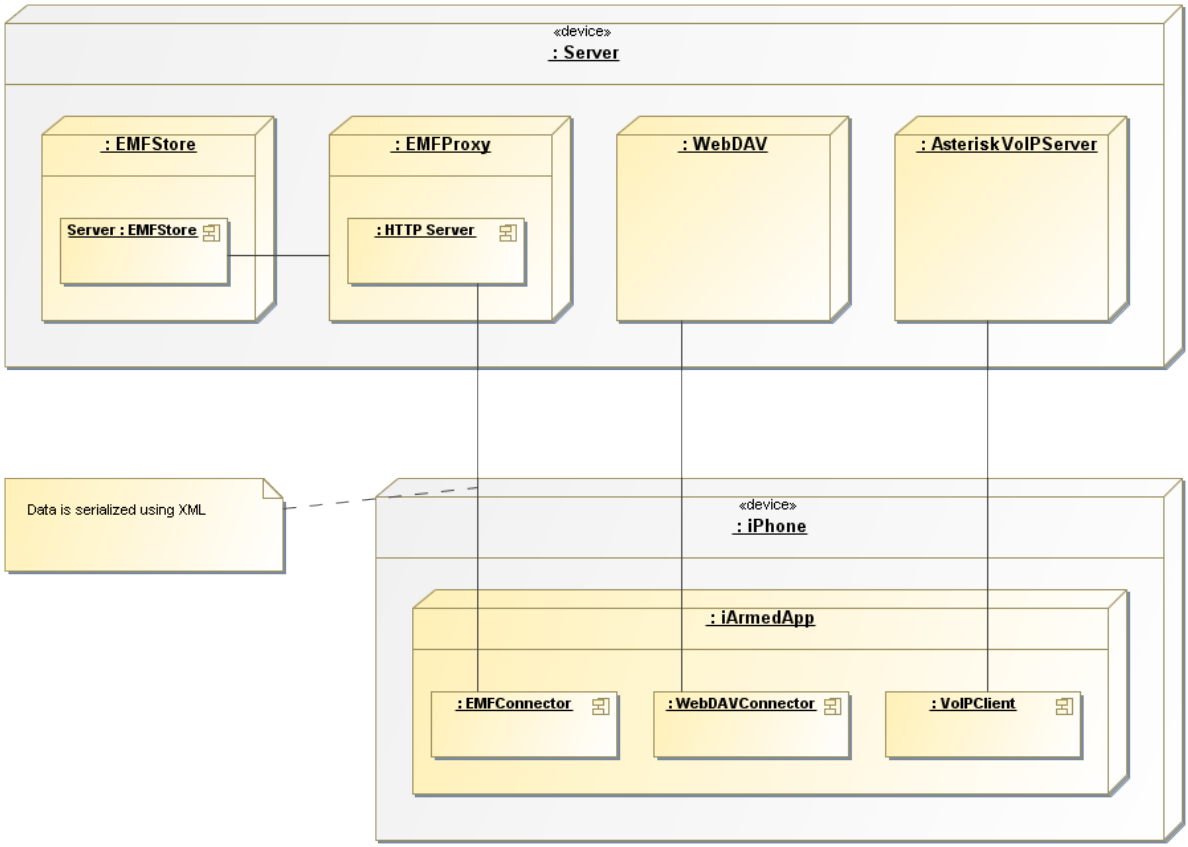
Server addresses:

EMF Store API
Url: https://ibmbruegge3.informatik.tu-muenchen.de:10443
User: super
Password: super

WebDAV
Url: http://dav.mrnobody.de
User: test
Password: test

VoIP
Url/IP: ibmbruegge3.informatik.tu-muenchen.de
User: iphone
Password: iphone

Component Diagram



Glossary

Maintenance in aviation:
Preventive maintenance: where equipment is maintained before break down occurs. This type of maintenance has many different variations and is subject of various researches to determine best and most efficient way to maintain equipment. Recent studies have shown that Preventive maintenance is effective in preventing age related failures of the equipment. For random failure patterns which amount to 80% of the failure patterns, condition monitoring proves to be effective.
Corrective maintenance: where equipment is maintained after break down. This maintenance is often most expensive because worn equipment can damage other parts and cause multiple damage.

Aircraft´s call sign: the aircraft´ s registration number (also called N-number in the U.S., or tail number). In this case, the call sign is spoken using the International Civil Aviation Organization (ICAO) phonetic alphabet. Aircraft registration numbers internationally follow the pattern of a country prefix, followed by a unique identifier made up of letters and numbers. For example, an aircraft registered as N978CP conducting a general aviation flight would use the call sign November-niner-seven-eight-Charlie-Papa.

EMF model: ...

Scenario: is a synthetic description of an event or series of actions and events.

Use case: is a description of a system's behavior as it responds to a request that originates from outside of that system. In other words, a use case describes "who" can do "what" with the system in question. The use case technique is used to capture a system's behavioral requirements by detailing scenario-driven threads through the functional requirements.

Use case diagram: is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Class diagram: is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

Object diagram: is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time. It focuses on some particular set of object instances and attributes, and the links between the instances. A correlated set of object diagrams provides insight into how an arbitrary view of a system is expected to evolve over time. Object diagrams are more concrete than class diagrams, and are often used to provide examples, or act as test cases for the class diagrams. Only those aspects of a model that are of current interest need be shown on an object diagram.

Component diagram: depicts how components are wired together to form larger components and or software systems. Components diagrams are used to illustrate the structure of arbitrarily complex systems.

Activity diagram: can be used to describe the business and operational step-by-step workflows of components in a system. It shows the overall flow of control.

CRC diagram: ...

SCD diagram: ...

SFD diagram: ...