**中图分类号：TN929**

论文编号：10006ZY1206216



硕士学位论文

**基于Android平台的电子飞行包（EFB）设计与实现**

作者姓名 盖玉杰

学科专业 计算机系统结构

指导教师 龙翔教授

培养院系 计算机学院

**Design and Implementation of Electronic Flight Bag (EFB) Based On Android Platform**

**A Dissertation Submitted for the Degree of Master**

**Candidate： Gai Yujie**

**Supervisor：** **Prof. Long Xiang**

School of Computer Science and Engineering

Beihang University,Beijing,China

**中图分类号： TN929**

**论文编号：10006ZY1206216**

硕 士 学 位 论 文

**基于Android平台的电子飞行包（EFB）**

**设计与实现**

作者姓名 盖玉杰 申请学位级别 工程硕士

指导教师姓名 龙翔 职 称 教授

学科专业 计算机系统结构 研究方向 Android应用开发

学习时间自 年 月 日 起至 年 月 日止

论文提交日期 年 月 日 论文答辩日期 年 月 日

学位授予单位 北京航空航天大学 学位授予日期 年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：        日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

摘 要

电子飞行包是实现电子化航空公司的一个重要步骤，是近几年民航业界内的研究热点之一，但我国在电子飞行包方面具有的独立自主知识产权的产品比较少。

本文以Android操作系统作为开发平台，设计与实现了面向通用航空的电子飞行包系统。该基于Android平台的电子飞行包系统采用了C/S模式，将驾驶员机舱内的纸质重要文档如机场的航空图表、机场的运行手册、规章标准以及机型手册和飞行手册等重要资料进行电子化存储、增量更新并展示，实现了电子检查单的交互式检查功能、电子飞行日志的自动生成功能、驾驶员机场滑跑位置预警功能和飞机起飞性能计算的功能。电子飞行包的所有功能都离不开地面系统数据的支持，本文还设计和实现了不同期航空图表资料的对比功能和导航数据库的制作、对比和维护的功能。

此外，本文的基于Android平台的电子飞行包系统还实现了将飞行计划中的航路点、情报区，天气等信息与GIS系统结合，以图形化的方式准确和方便的查阅飞机在飞行到某航路点和情报区的时候的油耗、风速和温度等信息的功能，这是大部分电子飞行包系统还欠缺的功能。

本文调研分析了国内外电子飞行包的研发情况，对基于Android平台的电子飞行包系统的各个功能部分的相关技术进行了对比分析，最终确定了基于Android平台的电子飞行包的各个功能部分的实现技术以及地面支持系统的开发框架，开发出了一套维护性强、使用方便的电子飞行包系统。

本文对基于Android平台的电子飞行包系统和它的地面支持系统进行了功能测试。测试表明:该基于Android平台的电子飞行包系统极大降低了飞行员在机舱内工作的负担，功能设计比较合理，贴近用户需求，用户体验良好，提升了飞机飞行安全水平，具有很好的市场前景和实用推广价值。

**关键词：**电子飞行包；移动终端；Android

**Abstract**

The electronic flight bag (EFB) is an essential step towards the realization of electronic airlines. Though it has gained great research focus of the civil aviation industry in the past few years, the electronic flight bag (EFB) has less products with independent intellectual property of China.

The paper designs and implements one EFB system for general aviation based on Android operating system. This system based on Android operating system, adopts C/S model to electronically store, incremental update and display the significant documents of flight deck such as aeronautical charts, operation manual, regulations and aircraft type manual and flight manual.

The system also achieves the function of interactive electronic checklists, the automatic generation of electronic flight log, forewarning of taking off and landing position for the pilot and the calculation of taking off performances. All the functions this system provides need the continual support of ground data system, therefore the paper designs and implements the comparison of different aeronautical charts and the creation, comparison and maintenance of navigation database.

Besides, the system based on Android operating system combines GIS system with the information of flight plan such as waypoints data, flight information region and weather condition which provides a graphical way for the pilot to accurately and conveniently check fuel consumption, wind speed, temperature and other information. Most electronic flight bag (EFB) products lacks this function.

This dissertation investigates the development of electronic fight bag (EFB) at home and abroad. The paper also performs a comparative analysis of the implementation of every part of the system which finally helps the determination of the implementation technology and framework to develop a powerful maintainable and user-friendly electronic flight bag (EFB).

The dissertation tests the function and performances of electronic fight bag (EFB) and ground data system. The tests shows that the system greatly eliminates the work burden of the pilots, closes to the requirements of the users, is of excellent users experience, can improve the level of flight safety, with good market prospects and practical promotional value.

**Key words:** electronic flight bag (EFB); mobile terminal; Android

目 录

[第一章 绪论 1](#_Toc405238815)

[1.1 电子飞行包（EFB）简介 1](#_Toc405238816)

[1.1.1 电子飞行包（EFB）的含义 1](#_Toc405238817)

[1.1.2 电子飞行包的分类 2](#_Toc405238818)

[1.1.3 电子飞行包的应用 5](#_Toc405238819)

[1.1.4 电子飞行包的作用 6](#_Toc405238820)

[1.2 研究背景及意义 7](#_Toc405238821)

[1.3 研究目标及内容 8](#_Toc405238822)

[1.4 本文的组织结构 9](#_Toc405238823)

[第二章 相关技术及原理分析 11](#_Toc405238824)

[2.1 Android系统应用程序的内存分配机制 11](#_Toc405238825)

[2.2 Android平台上的各种开源PDF阅读器 12](#_Toc405238826)

[2.2.1 Vudroid 12](#_Toc405238827)

[2.2.2 DroidReader 13](#_Toc405238828)

[2.2.3 APV pdf viewer 13](#_Toc405238829)

[2.2.4 Apdfviewer 14](#_Toc405238830)

[2.2.5 MuPDF 14](#_Toc405238831)

[2.3 数据增量更新 15](#_Toc405238832)

[2.4 电子飞行包航图资料和检查单 17](#_Toc405238833)

[2.5 Android平台上的GIS系统 18](#_Toc405238834)

[2.6 MFC和WPF 21](#_Toc405238835)

[2.6.1 MFC简介 22](#_Toc405238836)

[2.6.2 WPF简介 22](#_Toc405238837)

[2.6.3 MFC与WPF比较 23](#_Toc405238838)

[2.7 本章小结 24](#_Toc405238839)

[第三章 基于Android平台的电子飞行包（EFB）设计与实现 25](#_Toc405238840)

[3.1 电子飞行包总体设计 25](#_Toc405238841)

[3.2 阅读器设计 26](#_Toc405238842)

[3.3.1 MuPDF简介 26](#_Toc405238843)

[3.3.2 MuPDF多线程机制 26](#_Toc405238844)

[3.3.3 MuPDF使用 28](#_Toc405238845)

[3.3 数据增量更新 29](#_Toc405238846)

[3.3.1 技术设计 29](#_Toc405238847)

[3.3.2 增量更新设计 30](#_Toc405238848)

[3.4 机场飞行滑跑导航 30](#_Toc405238849)

[3.5 飞行计划导航 31](#_Toc405238850)

[3.5.1 飞行计划导航制作流程 32](#_Toc405238851)

[3.6 GIS切图 33](#_Toc405238852)

[3.7 性能计算设计与实现 35](#_Toc405238853)

[3.7.1 起飞性能数据库的获取 35](#_Toc405238854)

[3.7.2 将数据导入数据库 37](#_Toc405238855)

[3.8 导航数据库的设计与实现 38](#_Toc405238856)

[3.9 本章小结 44](#_Toc405238857)

[第四章 电子飞行包功能性测试 45](#_Toc405238858)

[4.1 测试环境部署 45](#_Toc405238859)

[4.1.1 测试数据介绍 45](#_Toc405238860)

[4.1.2 测试环境介绍 46](#_Toc405238861)

[4.2 测试过程及结果 47](#_Toc405238862)

[4.2.1 文档查阅测试 47](#_Toc405238863)

[4.2.2 起飞性能计算测试 48](#_Toc405238864)

[4.2.3 切图测试 49](#_Toc405238865)

[4.2.4 飞机滑跑测试 50](#_Toc405238866)

[4.2.5 飞行计划导航测试 51](#_Toc405238867)

[4.2.6 检查单测试 52](#_Toc405238868)

[4.4 本章小结 53](#_Toc405238869)

[总结与展望 54](#_Toc405238870)

[参考文献 56](#_Toc405238871)

[攻读硕士学位期间取得的学术成果 57](#_Toc405238872)

[致 谢 58](#_Toc405238873)

图 目

[图 1 AivlaSoft公司的EFB 2](#_Toc405238874)

[图 2 EFB硬件分类示意图 4](#_Toc405238875)

[图 3 智能手机搭载系统占用比 7](#_Toc405238876)

[图 4 移动开发者开发平台对比 8](#_Toc405238877)

[图 5 纸质资料电子化 8](#_Toc405238878)

[图 6 移动端EFB功用 9](#_Toc405238879)

[图 7 APV pdfviewer示意图 14](#_Toc405238880)

[图 8 Apdfviewer示意图 14](#_Toc405238881)

[图 9 OSMAND展示 19](#_Toc405238882)

[图 10 OsmDroid示意图 20](#_Toc405238883)

[图 11 EFB整体设计 26](#_Toc405238884)

[图 12 首都机场图 30](#_Toc405238885)

[图 13 飞行计划导航示意图 32](#_Toc405238886)

[图 14 飞行计划数据传输示意图 32](#_Toc405238887)

[图 15 PDF转瓦片地图 34](#_Toc405238888)

[图 16 DB2SQL使用示意图 36](#_Toc405238889)

[图 17 通过sql语句导入数据到sqlite数据库 38](#_Toc405238890)

[图 18 导航数据库软件界面 40](#_Toc405238891)

[图 19 导航数据库对比结果 41](#_Toc405238892)

[图 20 对比结果数据展示 42](#_Toc405238893)

[图 21 WPF数据绑定模型模型 43](#_Toc405238894)

[图 22 WPF数据绑定模式 44](#_Toc405238895)

[图 23 Android平台点在飞行包主页面 45](#_Toc405238896)

[图 24 电飞行包航图资料目录安排 46](#_Toc405238897)

[图 25 机场航图资料查询 47](#_Toc405238898)

[图 26 起飞机场资料分类展示 48](#_Toc405238899)

[图 27 起飞性能计算 49](#_Toc405238900)

[图 28 全国航路图展示 50](#_Toc405238901)

[图 29 飞机机场滑跑 51](#_Toc405238902)

[图 30 飞行计划导航示意图 52](#_Toc405238903)

[图 31 检查单使用示意图 53](#_Toc405238904)

表 目

[表 1 Android平台PDF阅读器对比 15](#_Toc405238905)

[表 2 GIS系统对比 19](#_Toc405238906)

[表 3 MuPDF第三方库列表 26](#_Toc405238907)

[表 4 EFB中调用MuPDF阅读器 28](#_Toc405238908)

[表 5 图层图标画法代码 33](#_Toc405238909)

[表 6 起飞性能数据库表设计 36](#_Toc405238910)

# 第一章 绪论

## 1.1 电子飞行包（EFB）简介

随着Android系统的迅猛发展和航空业的迅速发展，越来越多的航空公司开始认识到使用便携式电子设备，例如IPAD、Android和其他各种平板电脑，来查阅传统上使用的纸质检查单和参考文件，并逐步意识到便携式电子设备具有各种功能所带来的好处。各个航空公司开始使用便携式设备平台进行各种资料的电子化，研制一种名为电子飞行包（简称EFB）的便携设备，这种设备的使用减少了以往纸质资料的携带，给飞行员的查阅资料和记录航行带来了便利，更方便了资料的的频繁更新和管理。电子飞行包（EFB）几乎能够存储和显示飞行运行过程中所需要的所有的各种文件，比如公司运行手册、规章标准、检查单、最低的设备清单（MEL）、运行规范手册、各类航空图表手册和飞机起飞性能和降落性能分析手册 [[[1]](#endnote-2)]。

### 1.1.1 电子飞行包（EFB）的含义

电子飞行包（EFB）系统是飞机机长或者飞行驾驶员使用的一种飞行辅助工具，它既包含便于在飞机上携带的便携式硬件平台，也包含了在这种便携式硬件平台上实现功能的软件，它主要用于快速清晰的查阅和显示多种航行数据，执行一些基本的性能计算，以及动态获取一些与飞行、机组相关的航行信息。它专门设计用来彻底替代飞行员的传统工具包，其功能覆盖文档资料的查阅（如飞行计划的展示、运行手册、机型手册、航空图表、规章标准等）、性能的辅助计算（如飞机起飞性能计算、着陆性能计算等）、辅助的航图的查阅、飞行日志的记录、飞机机场滑跑的路径规划和预警、检查单的电子化检查等多个方面。它是一个能够大幅度减少飞行员工作量，同时又能够辅助飞机运营的功能强大的软硬件结合的系统设备。

根据美国联邦航空局的在咨询通告中对电子飞行包的定义，电子飞行包是在驾驶舱或者机舱内使用的一种电子显示设备系统。最简单的便携式电子飞行包系统( EFB)既可以和个人数字助理(PDA)兼容， 用于显示各种航行数据，进行各种飞行阶段准备时的计算和检查，或执行一些基本的计算(如飞机性能数据，油量计算等)。这些功能在传统上是采用纸介质参考文件的方式来完成的，或者使用航空公司的飞行签派部门提供给机组的数据资料。电子飞行包（EFB）系统的功能还可以包括一些集成在电子飞行包（EFB）系统内的数据库和应用，如预存各种空中需要使用的数据资料；复杂些的可以固定安装，等效于机上计算机和电子资料库(ELS)结合，与机上信息资源共享，成为无纸驾驶舱的有效工具。可以使用不同的技术、数据格式、通信方式实现EFB系统的信息显示，涉及的设备可以有辅助性能计算机(APC) 或膝上辅助性能计算机(LAPC)[[[2]](#endnote-3)]。AivlaSoft公司的EFB系统如图 1所示，它展示了电子飞行包的样式和基本功能。

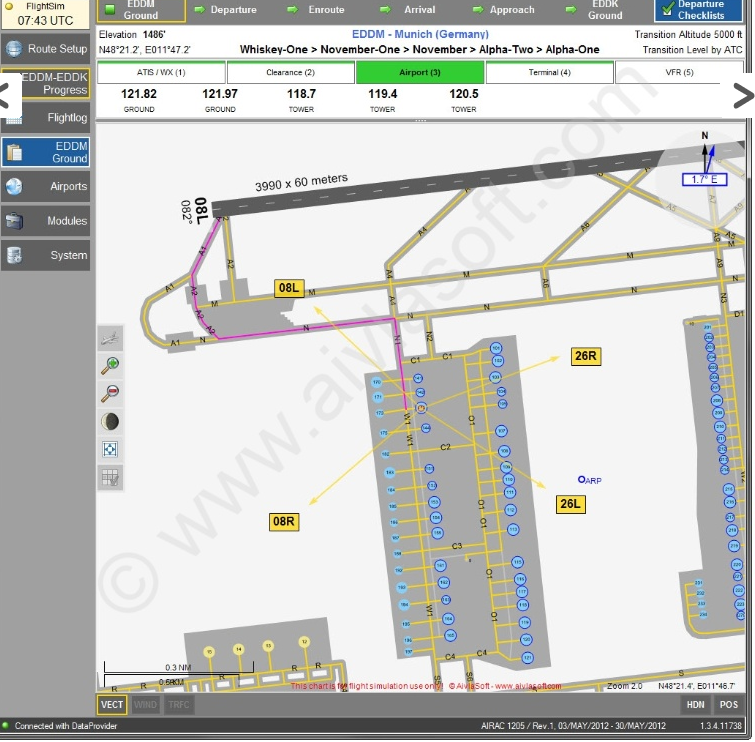


图 1 AivlaSoft公司的EFB

### 1.1.2 电子飞行包的分类

在民航局发布的资讯通告中(AC-121-FS-2009-31)，电子飞行包英文名称为Electronic Flight Bag(简称EFB)，定义为：包含用于支持一定功能的软硬件，用于驾驶舱或客舱的电子显示系统。EFB能显示多种航空信息数据或进行基本的计算( 如性能数据、燃油计算等) 。其中的一些功能传统上是使用纸质参考材料或是基于航空公司“飞行签派”向机组提供数据来完成的。EFB的功能范围可包括各种数据库和应用程序。电子飞行包（EFB）的显示可以使用多种技术、格式和通信形式。

电子飞行包从硬件和软件角度可以分别分成三级和三类。

（一） 硬件分类

(1)一级电子飞行包。

从电子飞行包系统运行使用的角度，1级电子飞行包EFB具有以下特征。

1. 商用成品计算机，用于航空器运行，目前常见的设备为IPAD，三星的平板电脑。

②此类电子飞行包不依附于航空器固定装置，不能用于飞行关键阶段。

所谓的飞行关键阶段即指对于飞行任务至关重要的起飞和进近着陆阶段，包括飞行过程中高度低于3000 m以下的除巡航以外的飞行阶段，包括起飞、离场、进场、进近、最后着陆、复飞等。

1. 在使用上，如果此类电子飞行包只使用A类应用程序，一般只用于文件查看和检查单的填写，不用于管理控制过程。

东航航空公司是国内第一家在驾驶舱内使用1级航空公司。东航航空公司经过华东地区管理局的正式批准，于2012年12月26日，在东航的A330机队进行1级电子飞行包的测试运行，也就是这一举措，使得东航成为中国国内首家获的批准在驾驶舱内使用1级电子飞行包（EFB）的航空公司。

(2)二级电子飞行包。

从电子飞行包（EFB）系统的系统运行与使用的角度，2级EFB是具有以下特征。

1. 商用成品计算机，用于航空器运行，目前常见的设备为IPAD、三星的平板电脑。
2. 在正常运行中与航空器固定装置相连接，可以用于飞行关键阶段和巡航阶段。
3. 由于在驾驶舱加装了支架，所以这类电子飞行包在航空器上的添加、拆卸或使用需要通过管理控制过程。
4. 组成设备/模块位于驾驶舱，机组容易取用，不使用工具就能拆除。

需要注意的是，由于是加装设备，二级电子飞行包的电源、数据线、天线等固定设备需要航空器适航审定部门给予批准。

(3)三级电子飞行包。

从系统运行使用角度，3级EFB是安装式设备，是飞机制造商生产时按照可续需求集成在驾驶舱的各显示系统中的，但其使用同样需要获得航空器适航审定部门批准。另外，3级EFB的硬件和安装的C类应用程序软件，除被用户修改的A 类和B类应用程序的软件除外，还应按照适航审定部门的相应规定通过合格审定。

目前几乎所有的主流大型客机如：A380B787等出厂时都已经安装3级EFB，但这些EFB的使用还需要局方的适航许可。

电子飞行包（EFB）的硬件分类图示如图 2所示：



图 2 EFB硬件分类示意图

（二） 软件分类

按软件可以分为三类，分别是A类应用软件、B类应用软件和C类应用软件。

(1)A类应用软件。

①可以在任何硬件等级的EFB上装载运行。

②必须由局方监察员评估其功能的适用性。

1. 不需要适航审定部门的设计批准。

④应在飞行员工作负荷降低的非飞行关键阶段使用，严禁在飞行关键阶段使用。局方批准的A 类EFB应用主要包括飞行手册、公司标准运行程序、运行规范、驾驶舱观察员简令卡等飞行前准备使用的手册或程序。

(2)B类应用软件。

①可以在任何硬件等级的EFB上装载运行。

②必须由局方监察员评估其功能的适用性。

③可能需要AEG 的评估。

④不需要适航审定部门的设计批准。

⑤可在所有飞行阶段使用。

(3)C类应用软件。

C 类应用需要适航审定部门的设计批准，用在A类和B类应用中的用户可修订的软件除外。C类应用的例子包括主要飞行显示。获得适航审定部门设计批准的一种方式是取得技术标准规定项目批准书(CTSOA)。C类包含了载重平衡和性能应用程序，是局方批准的特定飞机的应用程序。这些C类载重平衡和性能软件应用程序，被批准作为AFM或AFM补充件的一部分。

### 1.1.3 电子飞行包的应用

电子飞行包的应用十分广泛，我们选取了最基本的4种应用加以阐述。

(1)航图查看。

航图查看程序可以建立在平板电脑技术上，目前多数采用的是IPD为载体的平板电脑，JEPPESEN公司也发布了以三星平板电脑为载体的电子飞行包（EFB）系统。国内知名公司开发的EFB软件，它集成了国内航图和JEPPESEN公司制作的航图，并能够兼容FAA和公司自制的航图。支持中文、英文航图的显示与查阅，在安装了C类应用程序的EFB上，航图查看程序还能够将飞机的位置和航向描绘在机场图上。所有航图的使用可以做到全屏显示、缩放、滚屏和旋。除了航图查看的功能，电子飞行包还可以为飞行员提供搜索手册等便捷功能，从而减少查询步骤、减少查询时间， 从而增加飞行安全系数。此类程序，在所有级别的EFB上均可实现，而且是目前国内主流的EFB使用方向。

(2)机载性能工具(OPT)。

此类功能一般加装在3级EFB上，飞机性能模块提前装载数据包含：输入飞机的国籍、登记标志及对应的发动机型号；起飞性能分析、航路、进近和着陆、复飞等性能计算；动力装置的减推力性能计算和设置；成本指数模型；驾驶舱资源管理；主要飞行计划的更新；航线预演；载重平衡计算；故障保留单。

(3)电子视频监控。

飞行员可以利用飞机上的监控摄像头，通过EFB的信号处理功能，将视频信号显示在驾驶舱屏幕上，形成图像，使得机组可以对客舱人员活动情况、驾驶舱门情况、驾驶舱内人员活动情况和货舱内货物情况等飞机状态进行实时监控。这种功能一般只有在3级EFB上才可以实现。而且这种功能的实现，必须是由航空公司确定好摄像机的数量和位置之后，飞机供应商将按照要求，在所需部位对飞机的结构进行改造，适当加装视频监视设备，从而实现对飞机外部状态（机翼、发动机、各个操作面、飞机起落架等供做情况，以及对机翼积冰情况)进行实时拍摄、上传和监控；对飞机的各个内部航电、液压系统进行拍摄和记录；对飞机驾驶舱门、驾驶舱内情况进行实时拍摄和监控；对客舱活动情况进行实时拍摄和监控。

(4)电子检查单(包括各类起飞着陆检查单、应急检查单)。

电子检查单的主要作用是检查飞机状态，这种检查在遇到非正常或者在使用紧急检查单的时候非常重要，其中内置的操作程序主要是公司规定的飞行员参照完成的标准操作程序，目的是为了减少飞行员可在非标准状况下的思考时间，准确的完成最有利、最安全的操作程序，从而避免航空安全事故的发生，最大限度、最快速的帮助飞行员实现对飞机的控制。另外，电子检查单还具备自动检测和错误信息告警的提示功能。当出现错误时，系统会自动发出告警，提示机组注意。使用电子检查单来帮助机组更直观更形象地判断故障，更清晰更简明地处理故障1。

### 1.1.4 电子飞行包的作用

电子飞行包是实现电子化航空公司的一个重要步骤，它能给航空公司以及飞行员的飞行过程带来很多的好处。具体好处如下所示：

1. 减少驾驶舱内纸介质文件的数量，减少机组携带的资料重量，提高飞机组信息的检索与维护效率。
2. 提高机组携带资料的更新效率和准确率，有效降低航空公司纸介质文件的使用、维护成本，同时降低航空公司相关人员的工作负荷，提高工作效率，增强对飞行安全的保障水平与对机组提供服务的能力与水平。
3. 保障飞行所需资料，如航图、飞机性能资料、各项检查文件、气象资料、航行通告等实现及时、准确的内容更新，为飞机的运行安全服务。
4. 通过电子飞行包系统获取飞机发动机等重要部件的运行数据，并进行合理的数据管理与传输，使飞行机组及时获取关系飞行安全的重要信息，同时使航空公司等地面单位及时取得飞机的重要飞行数据，为保障飞行安全， 提高飞机运行效率服务。
5. 通过管理飞机系统采集的信息以及地面传输的信息，电子飞行包系统可及时、准确地计算起飞和着陆数据来降低燃油和飞机维护成本。
6. 通过管理飞机系统采集的信息以及地面传输的信息，可进行实时性能计算，在保障飞行安全的基础上提高飞机有效负载，提高飞机运行效益。
7. 及时获取航路、机场等区域的气象信息、航行通告信息，辅助飞行机组进行正确的形势判断，做出正确的操作决策。
8. 与地面单位，如空管与服务单位、航空公司相关运行控制与保障单位进行实时的双向数据通信，索取相关信息，同时向地面转发相关信息，为保障飞行安全和提高旅客服务质量服务。
9. 获取客舱视频信息，使飞行机组及时了解客舱状况。
10. 推动与提高空管与服务单位、机场相关单位、航空公司运行控制与安全保障单位等的信息化建设水平，为中国民航整体发展水平的提高起到积极作用2。

随着EFB系统不断发展，新的功能不断加入进来，EFB的功用也越来越多，它的好处也越来越多。

## 1.2 研究背景及意义

电子飞行包作为一种新型的电子显示系统，是航行信息管理和使用的一次重大变革。42%的航空公司已经实施电子飞行包，36%的航空公司通过平板电脑提供客舱服务，27%的航空公司通过平板电脑提供地面运营服务。预计2016年底，这一比例将超过70%。不过，大部分的航空公司的平台是基于IOS系统的IPAD，使用Android平台开发的电子飞行包（EFB）系统的航空公司比较少，一方面是由于Android平台上的电子飞行包系统比较少，另外一方面，国外的大公司比如JEPPESEN起初就是使用的IOS平台，推广的比较早。

虽然目前大部分的电子飞行包系统都是使用的基于IOS的，但是与此同时，Android系统近几年发展迅猛，全球约80%的智能手机都搭载Android系统，60%的平板电脑是Android系统。所有的计算设备中有60%也是Android系统，Windows位居第二，苹果是第三[[[3]](#endnote-4)]：如图 3所示：



图 3 智能手机搭载系统占用比

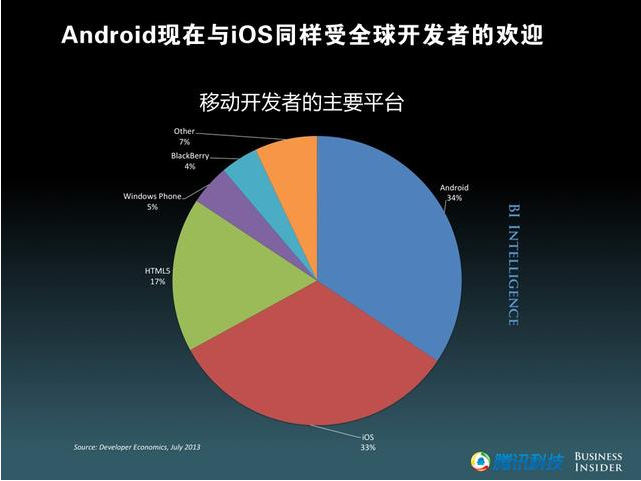


图 4 移动开发者开发平台对比

Android系统开源免费，各种生产基于Android系统设备的厂商如雨后春笋般诞生。Android系统的开发以及Android平台上的应用程序的开发吸引了越来越多的开发者。根据图 4可知，Android系统吸引了34%的开发者，比IOS的开发者还要多。

就是在这种发展背景下，开发基于Android平台的电子飞行包系统，是大势所趋。既可以减少系统使用的成本，又可以开发出具有自己知识产权的产品。

## 1.3 研究目标及内容

电子飞行包（EFB）系统最主要的是实现资料的电子化，如所图 5示：



图 5 纸质资料电子化

不过，随着Android系统的不断发展，功能的越来越强大，电子飞行包在实现了资料电子化的基础上，完全可以进行性能计算、飞机导航、通信等功能。

论文的主要研究目标是：

论文旨在设计和实现基于Android平台的电子飞行包，调研国内外电子飞行包（EFB）系统的研究现状，并在设计和实现了目前大部分电子飞行包（EFB）系统都有的航图资料的查阅、性能计算、电子化检查单的检查、文档是更新、飞行日志的保存和机场滑跑预警的功能外，结合现有的GIS系统的技术，根据飞行计划的内容信息，加入了飞机导航的功能，主要是将飞机从起飞机场到目的机场的过程中，经过的每一个航路点和情报区的天气等信息，画在GIS系统上，以图形化的方式快速查找和阅览每一个航路点和情报区的信息，如图 6所示。

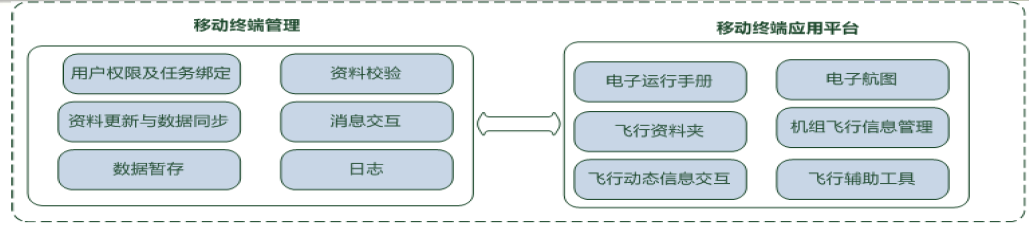


图 6 移动端EFB功用

论文的主要内容包括：

1. 调研分析电子飞行包（EFB）的概念、功能和作用。电子飞行包系统已经成了航空业的研究热点，人们借助现在良好的移动平台，完全可以利用数据的信息化和电子化，来替代以前的工作流程和工作方式，对电子飞行包概念的调研和分析，将目前电子飞行包上已经拥有的功能与现在的Android平台上的技术来实现，同时挖掘那些还没有实现过的需求，利用Android平台上的技术来解决。
2. 对基于Android平台的电子飞行包（EFB）系统的各个功能，调研分析可行的解决方案。文档的查阅需要Android平台上的阅读器，数据的增量更新，需要地面系统的支持，电子飞行包Android移动端也需要进行技术的支撑，机场滑跑的导航和飞行计划的导航需要GIS系统来实现。本文对以上部分的多种实现技术进行了分析，确定了最终系统的技术设计路线。
3. 设计实现一个基于Android平台的电子飞行包（EFB）系统。在Android系统强大的功能支撑下，本文在航图资料的查阅、性能计算、电子化检查单的检查、文档是更新、飞行日志的保存和机场滑跑预警的功能外，提供了资料的增量更新功能，节省了每次资料更新需要的时间，同时还对飞行计划文档进行分析整理，跟GIS系统进行结合，实现飞行过程中实时导航的功能。

## 1.4 本文的组织结构

本文共分为五章，具体内容安排如下：

第一章 绪论

首先对电子飞行包进行了简单的介绍，主要介绍了电子飞行包的含义，电子飞行包的硬件和软件的分类，电子飞行包的功用以及电子飞行的好处，然后介绍了基于Android平台的电子飞行包的研究背景和意义，最后介绍了本文的研究目标和内容等。

第二章 相关技术及原理分析

本章介绍Android系统中应用程序的内存分配机制，Android平台上的开源文档阅读器的对比分析，接着介绍了Android端和地面支持系统之间进行数据增量更新的SVN解决方案的测试分析，然后对电子飞行包系统各种航图等资料的进行分析，对检查单进行了整理，对Android平台上的GIS系统进行了分析总结，最后对地面导航数据库的系统开发框架MFC和WPF进行了对比。

第三章 基于Android平台的电子飞行包的设计与实现

本章详细介绍了基于Android平台的电子飞行包（EFB）的设计与实现。首先介绍了文档阅读器在系统整合和使用的方式，然后介绍了数据增量更新的设计与实现，电子化检查单检查功能的设计与实现。最后详细描述了电子飞行包（EFB）中机场滑行导航和飞行飞行过程中根据飞行计划导航的设计与实现。

第四章 基于Android平台的电子飞行包的测试

本章对基于Android平台的电子飞行包（EFB）系统进行了功能测试。功能测试结果表明，该基于Android平台的电子飞行包（EFB）系统能够对各种资料进行快速方便的查找，阅读器使用流畅，没有程序崩溃情况，数据增量更新快速准确无误，飞机机场滑跑导航，飞行日志的记录以及飞行过程中导航部分运行良好，能够方便飞行员正确驾驶。

第五章 总结与展望

对本文进行的研究实验和改进工作进行了总结，提出了本文的主要贡献，对后续工作的研究方案提出了设想和展望。

# 第二章 相关技术及原理分析

Android系统的迅猛发展，使得很多桌面和其他平台上技术能够很快的移植到Android平台上，电子飞行包系统中各个功能模块的设计与实现，可以从多种方案中选择，比如文档阅读就有Vudroid、MuPDF等开源项目可供选择、电子飞行包系统中Android平台上的GIS系统既有开源的瓦片和矢量图可供选择，也有商业的选择方案。而由于地面支持系统主要是运行在Windows平台上，所以主要选择微软的开发框架，比如MFC和WPF。本章对电子飞行包系统中的各个功能模块的设计方案进行了调研分析，对各种设计路线进行了简单的测试，经过分析和对比，确定了最终的技术方案。

## 2.1 Android系统应用程序的内存分配机制

电子飞行包系统中有GIS系统、大文档的查阅还有网络部分，特别是有两个200M左右的文档需要查阅。Android应用程序在应用过程中很容易出现OOM的情况，内存溢出出现的时机大部分是由于Bitmap解码的过程中，占用了大量的内存导致的。在电子飞行包系统中，如果加载的文档过于大，文档的页面转换成Bitmap后就会引起内存溢出的情况。所以在设计电子飞行包系统的时候，特别是在把阅读器加入到系统中的时候，需要合理的利用内存，防止出现内存泄露的情况。

Android系统主要有Native内存和Dalvik内存。Bitmap是直接在Native内存上进行分配的，程序中创建的Java对象是在Dalvik内存中分配的，Android系统中对应用程序使用内存的限制是Native内存和Dalvik内存的总和不能超过最大限制。如果两者占用的内存超过了Android系统分配给应用程序的能使用的内存的最大上限，那么就会出现内存溢出的情况。

需要注意的是，一旦内存分配给Java对象后，以后这块内存即使释放后，也只能供Java对象的使用[[[4]](#endnote-5)]，C代码是无法再使用这块内存的，所以如果Java对象突然占用了一个大块内存，那么C代码能够使用的内存就是分配给这个应用程序的内存的上限减去Java已经使用了的内存，如果剩下的内存特别的少，无法完成Bitmap内存的分配等操作的时候，就会出现内存溢出的情况。

解决内存溢出有两种方案，一种方案就是增大应用程序能够使用内存的上限，另外一种是合理的进行程序设计，使得程序能够合理的进行内存的回收再利用。

对于增大内存，Android系统中有largeHeap这个参数，查看Android设备的内存限制，可以再系统的/system/build.prop文件中，找到如所示两行：

dalvik.vm.heapsize=128m

dalvik.vm.heapgrowthlimit=64m

不同设备的值可能不一样，其中heapgrowthlimit就是一个普通应用程序的内存限制，而heapsize就是当largeHeap参数为True的时候，应用程序的内存限制。如果通过使用largeHeap参数为True，增大系统分配给程序的内存上限就可以解决内存溢出的问题，是最简便的方法，但是内存的大量占用也会带来垃圾回收器回收缓慢等问题。

通过程序的合理设计解决内存溢出的问题，主要是进行对象的合理创建和垃圾回收。

这中方案主要是在编程中解决，会增加程序设计的复杂度，同时如果数据比较大，比较多，如果每次都加载要使用的数据，可能会带来延迟，用户体验差。

## 2.2 Android平台上的各种开源PDF阅读器

电子飞行包（EFB）中的各种航图和航行资料都是pdf格式的文件，这就需要使用Android平台上的pdf阅读器进行阅读和查看。很多免费的pdf阅读器比如Adobe Reader和福昕阅读器，但是他们都是必须安装在Android系统以供使用。虽然在Android系统上，在查看航图和航行资料的时候，可以很容易的调用Android系统的接口，让用户选择使用哪一个本地的pdf阅读器进行查看，但是如果某一个Android上没有安装pdf阅读器，或者不小心删掉了之类的，就会导致无法查阅的情况，同时系统要求对pdf的操作除了简单的查看之外，还需要对pdf阅读器进行功能的定制，满足驾驶员查阅的特殊需求。

由于上面的需求，一款开源，功能强大的Android平台上的pdf阅读器必须集成在电子飞行包系统中。

本文对Android平台上的几款开源的pdf阅读器进行了调研和分析，综合分析，最后总选择一款pdf阅读器进行定制，并集成到电子飞行包系统中。

经过调研，Android平台上的几款pdf阅读器主要有：Vudroid、droidReader、apv、apdfviewer、mupdf。

### 2.2.1 Vudroid

VuDroid是一款基于DjvuDroid和PdfDroid的代码的djvu和pdf阅读器。它支持缩放，支持手势移动，能显示阅读的页数，打开大文件也没有问题。但是它是一部分一部分加载的，显得不流畅。新的VuDroid阅读器提供了DjvuDroid的特性，比如：侧滑放大缩小页面、快速切换横竖屏、修改了pdf崩溃的问题、滑动页面的时候显示当前页面的页数、文档的逐渐渲染以及更加有效的内存使用。但是使用的时候，可能会有些麻烦因为如果需要自己定制pdf阅读器的功能，就需要自己剥离pdf阅读器的部分，但是开发者底层用的Android Native Development Kit（NDK）开发，所以需要自己重新编译.So库文件，所以使用起来确实比较麻烦，不过，VuDroid使用起来确实比较流畅。鉴于开发者忙于其他项目，已经不再维护这个项目，所以VuDroid并不是电子飞行包（EFB）系统中的pdf阅读器的很好的选择。

### 2.2.2 DroidReader

DroidReader是一个PDF阅读应用程序，主要运行在谷歌的Android操作系统上。它使用了libjpeg、freetype和MuPDF这样的本地库来渲染页面。它同样是使用Android Native Development Kit（NDK）开发的。

在编译和安装过程中，发现了DroidReader的缺点：只能打开内容为图片的pdf；在解析过程中，少数的pdf的存在乱码；放大缩小很难操作，需要自己重新定制。

上面的问题主要是至少有7种变种PDF文档格式，所以要打开的PDF文档可能跟DroidReader能够支持的PDF文档的格式不兼容。

### 2.2.3 APV pdf viewer

APV采用的Native Library也是MuPDF。它是一个功能强大的阅读pdf的Android软件，它的主要功能在于阅读，高级版本还提供了书签导航功能[[[5]](#endnote-6)]。APV的缓存功能比较好，图片是切割成块的，只要原始显示图片不变，也就是说缩放级别固定，它们会把一张图片切割成固定大小的块，然后按照当前屏幕需要显示出一些块，所以在加载时，如果有些慢就会看到一块一块从黑到白的显示。再一旦加载成功后拖动，不会再调用底层C语言库切割，操作起来就会非常的流畅。经过测试和比较：APV刷新比Vudroid会稍微慢一点，同时APV也不支持多点触控。Vudroid读取文本格式的超强，20M的文字版的pdf文档， Vudroid都能顺畅阅读，Vudroid读取扫描版的或者里面有图片的PDF文档，速度稍慢；而apv读取扫描版的则顺畅一点，50M的扫描版也没有问题[[[6]](#endnote-7)]。APV pdf viewer示意图如图 7所示。



图 7 APV pdfviewer示意图

### 2.2.4 Apdfviewer

Apdfviewer在Android平台上的一个PDF文件阅读器，它是由Android Native Development Kit（NDK）开发，并且是整页加载模式中最快的一款，但是当它读取大文件读取的时候会比较慢，在Android系统上很容易得到程序长时间没有响应的提示。同时它只是一个PDF文档阅读器，并没有内建的文件管理器，而且必须通过Android的上的其他应用程序，比如Android自带的文件管理器或者其他任何能产生提示选择PDF阅读器的Intent的程序。Apdfviewer示意图如图 8所示。



图 8 Apdfviewer示意图

### 2.2.5 MuPDF

MuPDF是一种新的基于C的PDF解析库。MuPDF基于Fitz(一个新的和现代图形库)。Fitz显示树的核心:场景图的矢量图形、图像和文本的内容页面。

MuPDF也有一个API修改PDF文档的内部对象和写入PDF文件。例如，可以使用MuPDF库加密现存的PDF文件或重新安排页面。

MuPDF是一个轻量级的PDF和XPS查看器。MuPDF渲染器是专为高质量的抗锯齿图形。它呈现文本与指标和间距准确分数最高的一个像素内富达在屏幕上复制一个打印页面的外观[[7]](#endnote-8)。

MuPDF也比较小打开文档和翻页比较快同时功能强大。它支持PDF 1.7，具有透明度、加密、超链接、注释、搜索等功能。它还打开XPS和OpenXPS文档。 MuPDF是模块化的，所以任何开发者特别想添加的功能，都可以方便的加入进去。

自从MuPDF 1.2版本之后，它具有可选交互式特性，比如表单填写，Javascript和转换[[[8]](#endnote-9)]。

各种PDF阅读器的对比情况，如表 1所示：

表 1 Android平台PDF阅读器对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Vudroid | DroidReader | APV | Apdfviewer | MuPDF |
| 支持格式 | djvu，pdf | pdf | Pdf | pdf | xps，pdf |
| NDK开发 | 是 | 是 | 否 | 是 | 否 |
| 大文件读取 | 慢 | 慢 | 快 | 慢 | 快 |
| 兼容性 | 好 | 差 | 好 | 好 | 好 |

根据对比，MuPDF使用更方便，可扩展性更好，跨平台，打开文档速度快，使用流畅，同时MuPDF还有一个强大的维护团队，所以MuPDF是比较好的选择。

## 2.3 数据增量更新

数据的增量更新能够保证每次更新数据的时候，只会对所有文件中那些有过增删改的文件进行处理，而那些没有改变过的文档是不需要更新的。这样就会保证更新的时候，需要的网络流量少，更新时间短等。

SVN是Subversion的简称，是一个开放源代码的版本控制系统，它采用了分支管理系统，主要用来对代码版本的维护。正是SVN的这种版本维护的运行方式，使得它很适合帮助我们进行文档更新操作。SVN确实可以像一个时间机器一样，回到任意时刻的版本，查看任意两个时刻的版本变动，不止在协同开发中，即使在个人开发过程中，这种特性都是非常非常有用的，我曾经有过这种经历，对代码进行很多的修改，发现修改的想法根本是错误的，而这时我已经修改了多个文件，要想回退是非常纠结的事情，而现在可以使用SVN轻松做到这一点[[9]](#endnote-10)。

SVN是一个典型的C/S架构，服务端维护全部数据以及更改的记录，客户端维护一个本地的工作副本[[10]](#endnote-11)。

SVN与CVS优缺点分析：

所有的文档都显示SVN可以取代CVS，同时SVN的问题和缺点都被隐藏了。不幸的是，我们并不认为SVN是CVS的替代品，尽管很多缺陷都被修改了。更有甚者，它甚至让人重回VSS。CVS和SVN的比较类似于比较C++和Java。很明显CVS和SVN都远比SourceSafe强大的多，如同C++和Java比Basic强大的多。CVS代表了几乎代码控制系统的所有功能项，尽管有时他的实现并不很方便。SVN修正并添加了一些CVS并不拥有功能。例如，创建标志和分支dubious，你在编辑文件时其他人不会有任何通知。SVN并不是CVS的替代品，只是个不同的系统，类似于CVS。它有些特有的功能，足以作为采用它的理由。这些功能使他更适合于开发环境，例如对PowerBuilder。下面你可以找到两者的相对优势、劣势[[11]](#endnote-12)。

（1）存储类型格式

CVS是个基于RCS文件的版本控制系统。每个CVS文件都不过是普通的文件，加上一些额外信息。这些文件会简单的重复本地文件的树结构。因此，不必担心有什么数据损失，如果必要的话可以手工修改RCS文件。

SVN是基于关系数据库的(BerkleyDB)或一系列二进制文件的(FS\_FS)。一方面这解决了许多问题 (例如，并行读写共享文件)以及添加了许多新功能(例如运行时的事务特性)。然而另一方面，数据存储由此变得不透明。

（2）速度

CVS比较慢。

整体而言，由于架构实现的不同，SVN的确比CVS快很多。在网络上它只传输很少的信息并支持更多的离线模式的功能。但这也是有代价的，速度的代价就是巨大的存储（完全备份所有的工作文件）。

（3）标志和分支

SVN把采用标志和分支而抛弃了其他三件东西，实际上这意味着他们把这个概念替换为在档案库内部复制文件或目录以便保存日志。这样一来，无论标志创建还是分支创建都只是仓库内部的文件复制了。对分支而言：现在分支不过是在仓库内部的一个单独的目录而已了，不像早期还有些什么交错。对标志而言：已经不能对代码加标志了。在某种程度上说，SVN全文件编号补足了这个缺陷，SVN里整个仓库都有版本号，但不是针对单个文件。

（4）元数据

CVS只允许存储文件。

SVN允许一个文件有任意多的可命名属性，功能十分完全。

（5）文件类型

CVS最初是为文本文件存储而设计的。因此其他文件类型（二进制，统一码）文件的支持几乎没有，如需要的话则要有其他信息，并且客户端服务器端都要调整。

SVN会关心所有的文件类型，不需要你来手工操作。

（6）回滚

CVS允许任意的回滚，在任意一个已递交的版本上，尽管这要花些时间（所有的文件都要分别处理）。

SVN不允许递交后回滚。建议把版本库里好的状态版本加到末尾，覆盖掉损坏的版本。而损坏的版本无论如何也是会存在数据库里的。（SVN的滚回操作实际上是merge操作）。

（7）事务

CVS中的“零或一”事务原则根本没有实现。如果检入几个文件的话（加到服务器上），很有可能部分文件完成了，而另几个没有。作为一个潜规则，手工纠正这些并且对余下的文件 (而不是所有文件)一一重复检入。这样这些文件将在两阶段中被检入。SVN的确支持“零或一”事务原则，这是SVN的一大优势。

Open Android SVN(OASVN)是Android平台上的SVN软件，可以很好的跟SVN服务器进行良好的通信。

## 2.4 电子飞行包航图资料和检查单

电子飞行包系统中的有各种航图等资料，为了能够方便的供驾驶员查询，必须分门别类的将这些资料组织起来。

每个机场有一个唯一的四字代码，比如首都机场是ZBAA，那么就可以将首都机场的所有航图都放在ZBAA这个文件夹下。每个机场的航图资料主要有标准仪表进场图、标准仪表离场图、放油区图、机场图\_停机位置图、机场障碍物图\_精密进近地形图、进近图\_RNAV\_RNP\_RADAR\_GPS\_GNSS"、目视进近图、区域图\_空中走廊\_放油区图、仪表进近图、仪表进近图\_ILS、仪表进近图\_NDB、仪表进近图\_VOR、机场AIP、目视离场图，机型手册主要有FCOM、FCTM、MEL&CDL和QRH，此外还有运行手册和规章标准。这些资料需要合理安排目录存放位置方便用户快速检索。合理的安排文档资料的放置路径，以便电子飞行包系统可以方便的查找到需要打开和展示的资料。

检查单是飞机的飞行的各个阶段都需要检查的项目列表。它包含了起动前、起动后、起飞前、起飞后、进近、着陆、着陆后、停机、安全离机和起飞重心等各个项目的检查。可以将每一个需要检查的项以列表的形式展示出来，以便用户点击，表示检查。

## 2.5 Android平台上的GIS系统

地理信息系统（GIS ，geographic information system）是随着地理科学、计算机技术、遥感技术和信息科学的发展而发展起来的一个学科。在计算机发展史上，计算机辅助设计技术（CAD）的出现使人们可以用计算机处理象图形这样的数据，图形数据的标志之一就是图形元素有明确的位置坐标，不同图形之间有各种各样的拓扑关系。简单地说，拓扑关系指图形元素之间的空间位置和连接关系。简单的图形元素如点、线、多边形等；点有坐标（x， y）；线可以看成由无数点组成，线的位置就可以表示为一系列坐标对（x1， y1），（x2， y2），……（xn， yn）；平面上的多边形可以认为是由闭合曲线形成范围。图形元素之间有多种多样的相互关系，如一个点在一条线上或在一个多边形内，一条线穿过一个多边形等等。在实际应用中，一个地理信息系统要管理非常多、非常复杂的数据，可能有几万个多边形，几万条线，上万个点，还要计算和管理它们之间的各种复杂的空间关系[[[12]](#endnote-13)]。

目前大家比较常用的GIS系统有百度地图，谷歌地图和高德地图等。但是这些公司只是提供了使用它们的产品的SDK，地图必须使用它们本公司的地图，但是电子飞行包系统中的地图系统的地图需要地面支持系统的维护，不应该由第三方提供。所以需要使用能够自己制作的地图的GIS系统。

GIS中的地图有瓦片地图和矢量地图之分，瓦片地图是通过裁剪后的地图图片，以GIF、JPG、PNG等形式存储的，由于对多个比例尺的地图进行了裁剪，这样可以在显示的时候更快些。矢量地图就是画出来的点线区文件，它保存的时候占用的空间小，不失真。下面对比了SuperMap，ArcGIS，OSMAND和OSMDroid这四款GIS系统对比图如表 2所示。

表 2 GIS系统对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SuperMap | ArcGIS | OSMAND | OSMDroid |
| 地图格式 | 瓦片地图  矢量地图 | 瓦片地图  矢量地图 | 瓦片地图  矢量地图 | 瓦片地图 |
| 地图编辑软件 | 有 | 有 | 无 | 无 |
| 是否开源 | 否 | 否 | 是 | 是 |
| 地图格式 | smwu | shp | png、sm | png |

1. OSMAND(OSM Automated Navigation Directions)

OsmAnd(OSM Automated Navigation Directions)是一个基于免费的、广泛使用的、高质量的开放街道地图数据的地图导航应用。所有的地图数据可以以离线的方式保存在你的存储卡上。通过智能手机的GPS功能模块，OsmAnd为驾驶汽车、骑自行车和步行的人提供了地图导航和声音导引。它的主要特征有：能够离线和在线使用、声音导航、偏离路线自动导航、展示POIs（point of interests）、展示用户的位置和方向、地图来自Open Street Map保持数据更新、分享自己的位置、直接向OpenStreetMap进行反馈[[[13]](#endnote-14)]，而且它还是开源项目很适合自定义开发，只是欠缺桌面系统的支持。OSMAND的界面使用如图 9 OSMAND展示所示，



图 9 OSMAND展示[[[14]](#endnote-15)]

1. OSMDROID

Osmdroid是一个完全免费的Android自带的MapView的替代品。它还包含一个模块化的瓦片提供系统来支持大量的在线和离线瓦片资源，同时它还提供了图层支持，使用内置的图层来画图形、跟踪轨迹和绘制图标。它需要slf4j-android记录库，可以从<http://www.slf4j.org/android/>下载。使用的时候，只需要把jar包导入到工程之后，使用Osmdroid的MapView替换Android自带的Mapview即可[[[15]](#endnote-16)]。如下所示：

<org.osmdroid.views.MapView  
        android:id="@+id/mapview"  
        android:layout\_width="match\_parent"  
        android:layout\_height="match\_parent"  
        tilesource="Mapnik"  
        />

其中Mapnk是一个压缩文件，里面保存从OpenStreetMap上提取的瓦片地图，主要是根据用户画的区域和要保存的比例尺，按照一定的目录将png文件保存起来，以供地图加载使用。OsmDroid图展示如下：



图 10 OsmDroid示意图[[[16]](#endnote-17)]

（3）ArcGIS

ArcGIS是Esri公司集40余年地理信息系统（GIS）咨询和研发经验，奉献给用户的一套完整的GIS平台产品，具有强大的地图制作、空间数据管理、空间分析、空间信息整合、发布与共享的能力[[[17]](#endnote-18)]。

ArcGIS产品线为用户提供一个可伸缩的，全面的GIS平台。ArcObjects包含了大量的可编程组件，从细粒度的对象（例如单个的几何对象）到粗粒度的对象（例如与现有ArcMap文档交互的地图对象）涉及面极广，这些对象为开发者集成了全面的GIS功能。每一个使用ArcObjects建成的ArcGIS产品都为开发者提供了一个应用开发的容器，包括[桌面](http://baike.baidu.com/view/79807.htm)GIS（ArcGIS Desktop），嵌入式GIS（ArcGIS Engine）以及服务端GIS（ArcGIS Server）[[[18]](#endnote-19)]。

（4）SuperMap

SuperMap GIS是具有完全自主知识产权的大型地理信息系统软件平台。包括组件式GIS、开发平台、服务式GIS开发平台、嵌入式GIS开发平台、桌面GIS平台、导航应用开发平台以及相关的空间数据生产、加工和管理工具。使用SuperMap的桌面GIS平台可以开发出自己的地图，然后地图就可以在桌面端和移动端使用[[19]](#endnote-20)。

SuperMap和ArcGIS是商业软件，它会根据部署的设备的数量来收费，OSMAND和OSMDroid基于Open Street Map的数据的，它虽然免费，但是缺少强大的地面作图系统的支持，很难做到自己快速定制地图。同时由于目前能得到的地图都是SuperMap公司的桌面制图平台制作出来的，在地图格式之间转换的时候，可能会出现问题，所以选择SuperMap公司的产品来开发Android端的应用比较合适[[20]](#endnote-21)。

## 2.6 MFC和WPF

地面支持系统主要是导航数据库的制作和维护，导航数据库在飞行计划的制作等方面有着重要的作用。

地面系统主要是运行在windows操作系统上的，而Windows操作系统上的开发框架并不只有一种，现在比较流行的是MFC和WPF。

Microsoft Foundation Classes(MFC)是一个由微软公司提供的类库，是以C++类的形式封装了的Windows应用程序编程接口，并且包含一个应用程序框架，以减少应用程序开发人员的工作量。其中的类包含大量Windows句柄封装类和很多Windows的内建控件和组件的封装类。

Windows Presentation Foundation(WPF)代表了用户界面技术的一个重要进步，是微软Vista操作系统的三大核心开发库之一，其主要负责的是图形显示，所以叫Presentation(呈现)[[[21]](#endnote-22)]。

本文对他们进行了调研分析，以便确定桌面系统的开发框架。

### 2.6.1 MFC简介

MFC是一种应用程序框架，该类库提供一组通用的可重用的类库供开发人员使用。它的大部分类均从CObject直接或间接派生，它的部分类为MFC与活动模板库通用，可以在Win32应用程序中单独包 含并使用这些类。

MFC用于在C++环境下编写应用程序的一个框架和引擎，VC++ 是Windows下开发人员使用的C++专业软件开发平台，MFC就是挂在 它之上的一个辅助软件开发包，它是一个非外挂式的软件包、类库，不过MFC类是微软为VC++专配的。

MFC是一种软件编程的规范，但不是一种程序开发语言本身，它是微软对程序接口函数的专用C++封装，这种结合使用户在Windows 下应用程序的开发变得方便。并且微软做了大量的工作，隐藏了程序 开发人员在Windows下用C++／MFC编制软件时的大量内节，进一步简化了程序开发过程。

MFC提供的类绝大部分用来进行界面开发，关联一个窗口的动作。但它不仅仅只是一个功能单纯的界面开发系统而已。它提供的类中也有很多类的作用不是用作界面类的作用，而是一些在Windows中实现内部处理的类。

MFC是微软封装了的应用程序编程接口，这方便了程序员的开发工作。因为应用程序编程接口函数非常多，且名称很乱。如果从零构架开始编程，一个窗口动辄就是上百行的代码，但是MFC将传统的应 用程序编程接口进行了分类封装，并且为用户创建了程序的一般框架，从而大大的方便了程序开发工作。

### 2.6.2 WPF简介

WPF是基于微软多媒体编程接口DirectX的，它是一种新的图形引擎，相应增加了很多新的功能。比如它强大的2D和3D性能和对Aero图形引擎的支持(Aero是专门为3D桌面开发的引擎，可以让桌面实现3D翻转)。WPF的重要性主要体现在以下两个方面[[22]](#endnote-23)。

首先，WPF架构提供了一种全新的开发模式，即显示界面设计和代码编写是可以分离的。这种新形式使得设计可视化的用户界面的设计者和编写内部核心代码的开发人员可以并行工作。它所带来的好处体现在以下四个方面。

①将此二者分开，有助于程序编写者将注意力集中于程序设计本身，从而设计

出更加优良的代码。

②WPF的这种新型设计方式使得一些商业的设计者们能够更好的从事界面设计工作，并集成一些实用的功能，而购买的核心代码能够独立的完成内部的一些关键逻辑但不干扰设计过程。

③它可以为不同的用户采用定制的界面而保持核心的逻辑不变。开发者们创建程序逻辑，然后为不同的用户使用不同的界面并贴上他们的个性标志或定制图元等。使得软件产品风格多样化和个性化。WPF 的界面设计使用的是XAML(全称eXtensible Application Markup Language，中文名称是可扩展应用程序标记语言)，这是一种基于XML 的语言。

其次，WPF使用一种基于向量的绘制引擎，这就有别于且优于之前的基于光栅的绘制引擎的Windows。具体地说，光栅绘制引擎通过在屏幕上绘制象素点来绘制表面，象素只是点。所以，如果屏幕分辨率(DPI，每英寸点)增加了，它的效果就会变差。但是现在屏幕都采用超高DPI的分辨率，字体必须使用成百上千个DPI来达到一定的浏览效果。管理这些基于光栅的图像的点将会消耗大量的处理能力，所以很浪费。而采用基于向量的方式来替代象素，在一个可扩展的坐标系里绘制字体和其他线型，使得它们可以独立于DPI。同时，向量图也使得一些变换变得更加方便和易于操作，比如3D、旋转和动画。

### 2.6.3 MFC与WPF比较

在代码生成方面，MFC生成本机代码，自足很快。可是，消息循环却又减缓了界面显示速度。WPF是一种新的模型，直接访问驱动程序，加快了运行速度。不过，这种模型需要更好的显卡，所以硬件要求更高些。

在开发效率方面，WPF要优于MFC。尽管MFC开发界面执行效率高但是开发效率低，时间和开发效率往往能决定项目开发的成败，所以WPF的优势明显。

在开发成本方面，MFC高于WPF。用MFC开发成本太高，对开发者能力要求更高，作为客户当然希望开发的费用越少越好，开发者当然希望钱赚得越多越好。所以在成本方面WPF优于MFC。

在界面执行效率方面，MFC与WPF基本持平。这是因为现在计算机硬件的性能提高很快，多核CPU已经普及，所以它们的差距已越来越小。

在美观方面，WPF很容易超越MFC。因为MFC中界面开发通常 是需要程序员自己来完成的，但是WPF的界面开发因为可以与程序编写分开，所以可以让专业的美工人员参与设计，在美学上就会更甚一筹，再加上WPF出色的显示引擎使得显示效果更佳绚丽，这就使得WPF的美观性方面大大优于MFC软件。而且在MFC下要开发出一个华丽的用户界面用户所需要使用的控件在商业开发上是要收费的，这对于商业开发是不经济的。而WPF很容易就可以做出像Vista那样的用户界面特效，优劣立现。

在内存使用方面，当然WPF要比MFC更加消耗内存。但是随着计算机硬件的性能提高WPF这个缺点会被忽略。

此外，在最新版的WPF编程平台Visual Studio2013中，当用户在代码界面进行代码编写时，该平台系统会不断有相应的代码提示自动跳出来，非常的有助于启发思路和提高效率。甚至，在用户只构思出了所想要实现的功能，但是对所需要用到的函数或者方法一无所知时，用户可以仅凭相应功能的英文单词来引导系统联想出相应的函数或者方法，这一人性化和智能化的功能为编程者大大节省了思考的时间，节约了脑力和精力。这个特性也是MFC编程平台所无法比拟的。

综合以上各个方面两者之间的比较可知：WPF在大多数方面要明显优于MFC。

随着WPF的出现，MFC一些缺点日益突出。比如，入门门槛比较高，而且同样完成一个任务代码量相对较多。而原有的优势如运行速度快等，也因为个人电脑的运算速度增加而显得不那么突出。但是MFC仍有其独到之处：譬如，面对底层程序，它能很轻松的与Windows、应用程序编程接口或驱动程序结合，即在自己的代码中直接使用应用程序编程接口函数。而应用程序编程接口和驱动程序的资料都是以C 语言为基础的，这使得MFC程序员能够更轻松的使用Windows应用程序编程接口。MFC是一种入门困难，但是扩展学习却很轻松的语言框架。如果既限于某一领域的话MFC毫无优势可言，但是如果开发一个新的领域的应用程序或者该程序涉及多个应用领域的话，可减少重复学习的频率和难度，MFC的优势会立刻显现出来。

## 2.7 本章小结

本章对基于Android平台的电子飞行包系统中各个功能模块的实现技术进行了调研和分析，主要是对Android应用程序内存分配的分析、对多种文档阅读器的分析和比较、对数据增量SVN的介绍、对电子飞行系统中各种资料的分析总结，对多种GIS系统的分析比较还有对地面支持系统框架MFC和WPF的介绍和对比。

# 第三章 基于Android平台的电子飞行包（EFB）设计与实现

本文的基于Android平台的电子飞行包设计与实现，包括Android端的系统的设计与实现和地面支持系统的设计与实现。Android端电子飞行包主要包括文档的快速阅览、文档版本的维护、机场滑跑、飞机飞行导航和电子化检查单。地面支持系统主要是导航数据库的制作和维护。

## 3.1 电子飞行包总体设计

电子飞行包（EFB）系统是包含移动端和地面端的一套系统。地面系统提供数据支持，移动端负责数据接收和使用。

在地面支持系统中，需要对各个机场的航图数据进行维护，比如机场跑道的修正，机场扩建，机场航路的变化等，都需要及时的更新该机场的机场图上的信息；航图和航公公司的公司资料也要保持更新；性能计算数据库也要在地面系统中通过添加新数据或者修正老数据，保证性能计算数据中的数据的可靠性。在地面支持系统中，飞行计划的发送，已经导航数据库的生成和发送是非常重要的，它指引了某次飞行的过程，对这个飞行过程中的各个航路点和情报区的数据进行了电子化的同时，也进行了地图化展示。方便飞行员的使用。

在Android客户端，主要负责数据的接收和使用。对于比较大，比较多的数据，可以在局域网内进行更新，这样可以减少传输数据产生的流量带来的费用同时传输速度比较快，可以节省传输时间。而对于飞行计划、性能数据以及航行日志等这些数据量比较小，更新比较频繁的数据，就需要使用3G网络或者蓝牙进行解决。

电子飞行包（EFB）系统的使用流程如图 11所示：

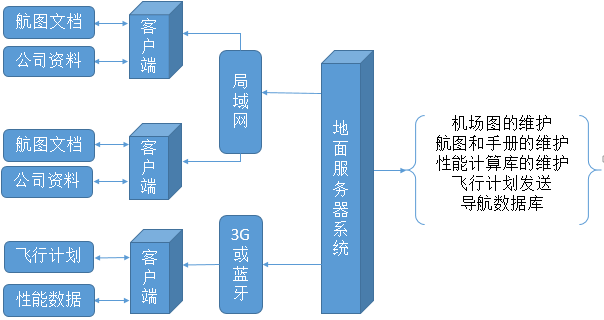


图 11 EFB整体设计

## 3.2 阅读器设计

Android平台上最终使用了MuPDF阅读器来进行文档的阅览，MuPDF阅读器是跨平台的，他不仅支持Android平台，而且还是windows phone和ios平台，可以在平台之间很好的切换和移植。下面主要对MuPDF阅读器的代码结构进行了分析，对MuPDF阅读器集成到电子飞行包系统中的过程进行了解析，对MuPDF在代码中的使用过程进行了解析。

### 3.3.1 MuPDF简介

MuPDF底层是使用了C语言库进行实现的，它使用第三方库列表如表 3所示：

表 3 MuPDF第三方库列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Library | Version | Function | License | URL |
| freetype | 2.4.11 | 字体缩放渲染 | Freetyp许可 | http://www.freetype.org/ |
| jpeg | 9.0 | jpeg解码 | 免费许可 | http://www.ijg.org/ |
| openjpeg | 2.0.0 | Jpeg 2000解码 | BSD类型 | http://www.openjpeg.org/ |
| zlib | 1.2.7 | 压缩 | zlib许可 | http://www.zlib.net/ |
| (Optional)  v8 | 3.9 | JavaScript解码 | BSD | http://code.google.com/p/v8/ |
| curl | 7.31.0 | Http数据传输 | MIT类型 | http://curl.haxx.se/ |

### 3.3.2 MuPDF多线程机制

MuPDF可以被构建成一个多线程应用程序并且并不需要让底层的库知道任何线程。如果在一个线程里打开一个文档，然后这个线程一直存在作为“服务器”的请求页面和并且给需要页面的其他线程进行渲染页面，这样的话，底层的库只会被这个作为服务器的线程调用。

其他线程仍然可以被用来处理UI请求等，但对于MuPDF而言只有以单线程的方式来使用。在这种线程任务分配的情况下，MuPDF没有线程问题了，它可以安全的使用而没有任何的阻塞，就像上面讲的一样。

本节将试图解释如何在更复杂的情况下使用MuPDF，比如在一个应用程序中从多个线程中同时访问MuPDF的库。

MuPDF提供了一组锁定函数来方便的供开发人员调用。它使用这些函数来对多个线程中可能产生的访问冲突采取了互斥操作。

应遵循以下简单的规则，以确保多线程操作顺利进行：

（1）不同线程不能使用相同的上下文来对MuPDF进行没有同步的调用。

大部分情况是只是需要简单的对每一个线程使用一个不同的上下文，所以只需要你在创建一个线程的时候，同时创建一个新的程序运行上下文就可以了。

（2）文档绑定到创建它的程序运行上下文。

所有后续对文档的访问隐式地使用相同的上下文；这意味着只有一个线程可以能够立刻访问文档。这并不意味着文档只能从一个线程使用，尽管在很多情况下，这是最简单的整体结构。

（3）任何设备绑定到上下文。

所有后续设备的使用都隐式地使用创建MuPDF对象的上下文；这意味着，如果一个设备已经用来打开了pdf文档，那么它用来创建它的上下文，跟创建文档的上下文是一样的。这并不意味着这个设备只能用一个线程来访问和使用，尽管在很多情况下这是最简单的情形。

那么一个多线程的MuPDF使用例子跟一个非多线程的使用例子有什么区别呢？

首先，当我们创建第一个上下文的时候，我们称之为fz\_new\_context，但第二个参数应该指向一组锁定函数的指针。

被调用的代码应该提供FZ\_LOCK\_MAX互斥，在提供的机构体里面，会有结构体指针和锁定编号（0<=i<=FZ\_LOCK\_MAX），MuPDF通过调用这些结构体指针和锁定编号就能够锁定/被锁定。这些互斥锁可以安全地递归和非递归因为MuPDF使用的是非递归调用风格。

在随后的上下文创建过程中，用户不应该再调用fz\_new\_context(这将无法分享重要资源比如存储和glyphcache)，但应该调用fz\_clone\_context。任何一个被克隆的上下文，都可以按照统一方式被fz\_free\_context释放。

为了打开一个文档，按照统一方式，通过传递程序上下文和文件名来调用fz\_open\_document；这个上下文会绑定到文档。所有以后对文档的调用都会在内部使用这个上下文。

因此一次只有一个线程可以执行一定的操作，如获取一个页面，或渲染页面显示列表。然而，一旦取得了显示列表，它就可以从任何其他线程(甚至同时从多个线程同时渲染)。

这意味着一个实现者有两个基本选择来构建一个应用程序使用MuPDF多线程模式。要么他可以构造这样一个指定线程打开文档，然后充当“服务器”为其他线程创建显示列表来渲染，或者他可以添加自己的互斥来调用mupdf使用文档。从长期来看前者可能是更有效的方法[[[23]](#endnote-24)]。

### 3.3.3 MuPDF使用

Mupdf阅读器的源文件包含两部分：一部分是Java语言部分，它主要是负责创建Android平台上的UI以及调用底层图形库；另外一部分就是图形库等的C代码部分，已经编译好，可以直接使用。

对于.so文件，直接放在Android工程的libs文件夹下，Java部分则将整个包放在Android工程的下面。

（1）Java部分解析

MuPDF中展示PDF的类是MuPDFActivity.java，在界面上获取到要打开的pdf文档后，生成相应PDF文档的文件路径，然后在生成的Intent中传递url进去就可以了。代码如表 4所示。

表 4 EFB中调用MuPDF阅读器

|  |  |
| --- | --- |
| 1 | *Uri*  uri =*Uri*.parse("/mnt/extSdCard/EFB/Airport/"+*HomePage*.versionlog+"/" |
| 2 | +*HomePage*.qiFeiJiChang+"/"+\_pdfNamesposition); |
| 3 | *Intent* intent = new *Intent*(this，*MuPDFActivity*.class); |
| 4 | intent.setAction(*Intent*.ACTION\_VIEW); |
| 5 | intent.setData(uri); |
| 6 | startActivity(intent); |

（2）内存泄露

MuPDF在集成的过程中出现了内存泄漏的问题，对于只有一张航图或者只有十几页或者上百页的文档而言，MuPDF完全可以快速的渲染页面，并且在页面的浏览和切换的过程中使用和流畅，但是对于200M左右的PDF文档，打开文件依然很快，但是当不断的翻页的时候，会出现内存溢出错误，导致整个程序崩溃退出。

对于MuPDF内存溢出的问题，解决方案有两种:一种是增大内存；另外还有一种就是在文档翻页的时候，只是预取当前页前后几页的内容，保证不会把整个文档都加载进来，这样可能会出现，跳转到某一页，等待页面渲染的问题。

## 3.3 数据增量更新

数据的增量更新的方法是使用版本维护。电子飞行包（EFB）系统中中航图资料和航空公司的资料更改是很频繁的，又由于所有文件的特别的大，如果每次更新都是全部文件的覆盖会特别的耗时，所以在基于Android的电子飞行包（EFB）系统中，使用了SVN的方式对资料进行维护，实现增量更新。

具体是实现方式是，Android端电子飞行包系统使用了OASVN，一款Android端的SVN客户端，地面支持系统中使用了VisualSVN服务器和TortoiseSVN的客户端管理软件。

### 3.3.1 技术设计

选用Visual SVN Server而不是Subversion的原因是因为Subversion Server比较复杂。Subversion一般还要安装配置Apache和trac，有时在版本兼容与配置上面，也要比较多的时间[[[24]](#endnote-25)]，同时Visual SVN Server是服务端，集成了Subversion、Apache和用户及权限管理，而且它是完全免费的，所以我们选择VisualSVN Server进行资料的管理和维护。

TortoiseSVN是Subversion版本控制系统的一个免费开源客户端。它的特性有外壳集成、重载图标和Subversion命令的简便访问。

1. 外壳集成。

TorToiseSVN与Windows外壳（例如资源管理器）无缝集成，你可以保持在熟悉的工具上工作，不需要在每次使用版本控制功能时切换应用程序。

1. 重载图标。

每个版本控制的文件和目录的状态使用小的重载图标表示，可以让你立刻看出工作副本的状态。

1. Subversion命令的简便访问

所有的Subversion命令存在于资源管理器的右键菜单，TortoiseSVN在那里添加子菜单[[[25]](#endnote-26)]。

OASVN是第一个功能性的SVN(Subversion)Android客户端，它正确使用SVN协议。 适用于http://、https://和svn://存储库。OASVN是一个免费和开源的GPL下许可的软件。

最新版本的OASVN支持以下SVN操作：检验、更新、提交、添加(文件添加到工作拷贝目录提交自动添加)、输出(允许从远程仓库检出)、清理、恢复等功能。

### 3.3.2 增量更新设计

首先需要建立好Visual SVN服务器上的仓库，将各自的文件，分别上传到仓库上面去，在客户端需要设置好仓库的正确地址。

## 3.4 机场飞行滑跑导航

飞机在机场的滑跑主要是飞机在机场滑行的过程中，根据控中心发送的命令将该飞机在机场的滑行路径画出来。下图首都机场为例，当驾驶员被指挥沿着跑道T3、跑道A1和跑道F3进行滑跑的时候，需要在地图上画出相应的路线和飞机所在的位置。如图 12 首都机场图的首都机场跑道路线图。

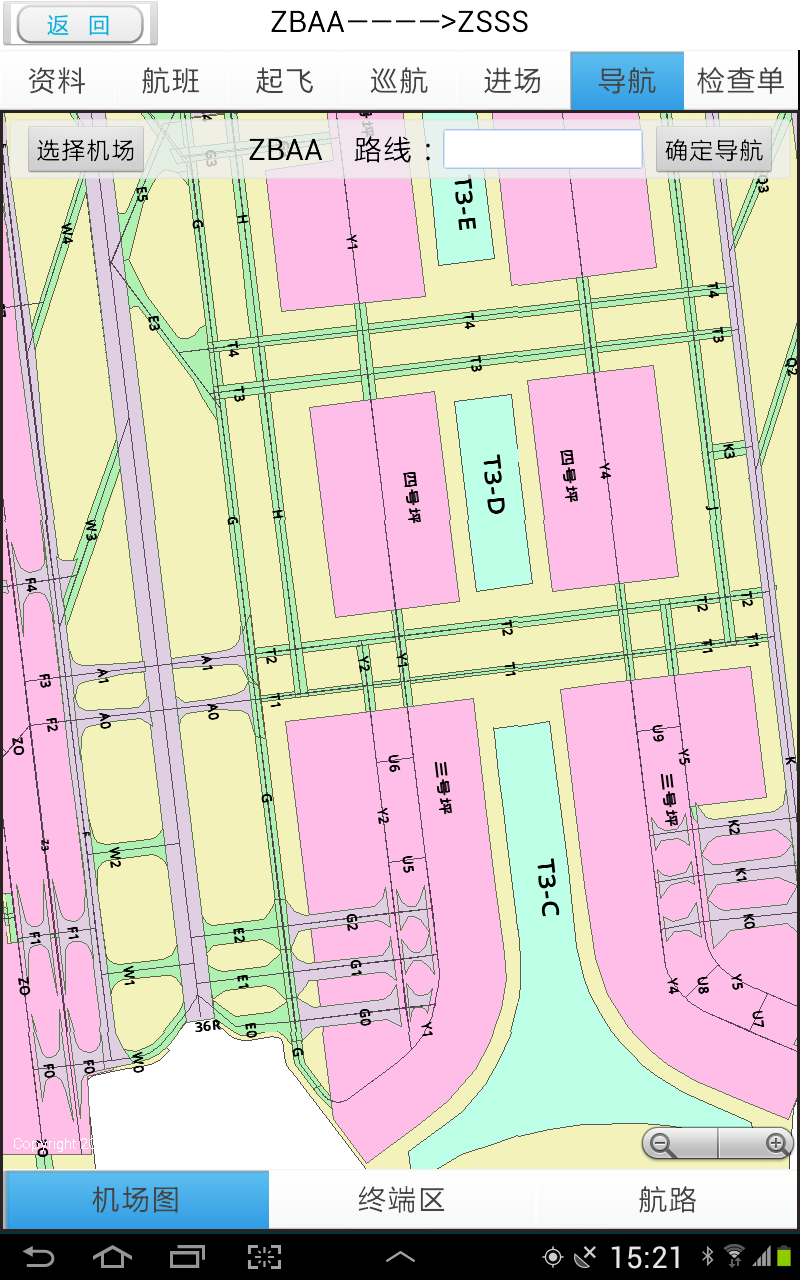


图 12 首都机场图

获取飞机的位置使用了GPS定位，Android系统中有比较完备的接口，主要是通过LocationManager来判断Android设备的GPS模块是否已经打开，如果已经打开，那么可以直接进行GPS信息的获取，如果没有打开，那么会跳转到GPS信息打开页面。同时需要使用SuperMap的Android端GIS系统在他的图层上进行绘制，主要步骤是首先获取地图的MapControl，然后获取拥有跑道的图层，从这个图层上获取数据矢量集，然后根据输入的跑道的编号进行数据信息的查询，得到每条跑道的记录集合，然后把线路画在另外的一个图层上。

## 飞行计划导航

飞行计划是指在每次航班飞行前，都应当根据具体的气象资料、航行情报、航空器性能、空中领航和航行规则计算出确定可带的商载以及完成本次航班飞行所需的飞行时间和燃油油量，其目的是为了保证航班飞行的安全性和提高运营的经济性。

它的主要内容是确定最大起飞重量和最大着陆重量，以便尽可能提高商载能力。根据需要选定飞行剖面中各段的速度和高度，以便节约燃油。计算各飞行阶段所需要的燃油量和时间，并由此得到该航班飞行所需的总燃油量和总时间。这是飞行计划的主要内容。给出的有关航路资料，包括航路点的位置、经纬度、导航设备的电台频率、呼号、各航段的航路代号、各航段的航向、距离等。如下图所示是北京飞往九寨的飞行计划的航路点的信息和SuperMap GIS地图进行结合的样子。如图 13所示是北京飞往九寨的飞行计划导航示意图

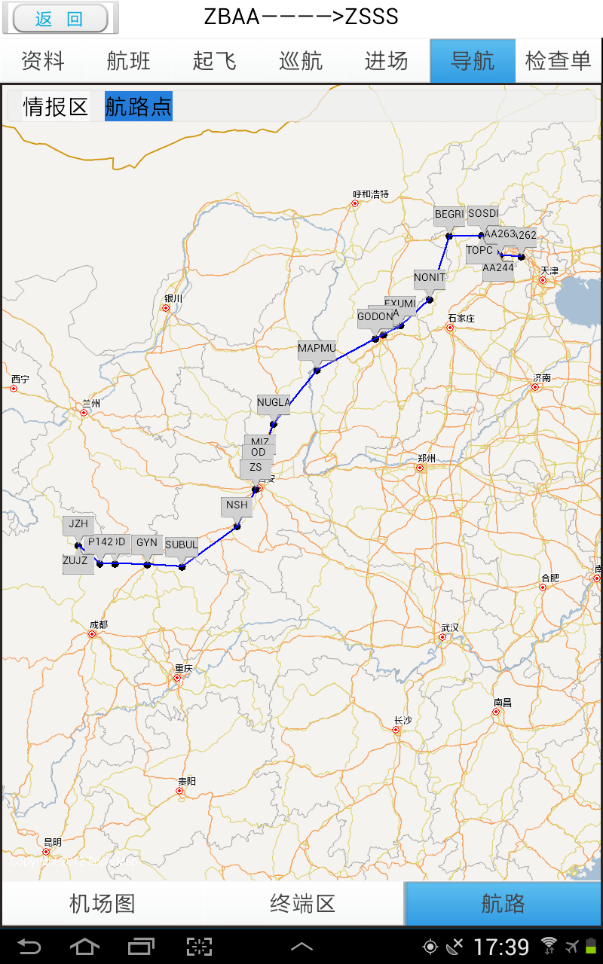


图 13 飞行计划导航示意图

### 3.5.1 飞行计划导航制作流程

飞行计划导航部分采用了桌面系统对飞行计划进行分析，生成飞行计划数据库，然后通过3G网络或者蓝牙将数据传送到Android平台上，数据库会在导航部分，跟SuperMap GIS系统进行结合，以图形化的方式快速的将飞行计划展示出来。制作流程图如图 14下所示：

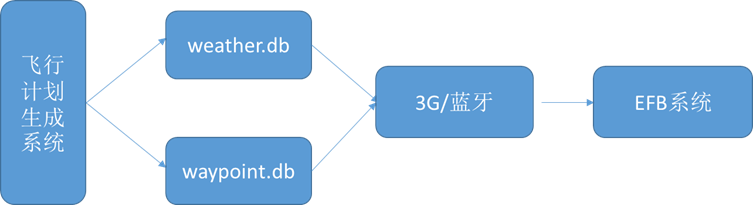


图 14 飞行计划数据传输示意图

飞行计划导航部分主要是使用到了GIS定位系统以及GIS地图的图层上画出新的标记的功能。所以图标的图画，可以使用SuperMap函数库中的Callout对象，而GIS地图上的所有点、线以及矩阵等都是靠点来画出来的。所有画在TrackingLayer上的图形，都有一个唯一的标识。可以根据这个标识来进行删除图形的操作。

航路点的代码设计如表 5所示：

表 5 图层图标画法代码

|  |
| --- |
| CallOut callout = new CallOut(HangLu.this);  callout.setContentView(calloutLayout); // 设置显示内容  callout.setCustomize(true);  callout.setLocation(qingbaoquPoint.getItem(i).getX()， qingbaoquPoint.getItem(i).getY());// 设置显示位置  mMapView.addCallout(callout，qingbaoqui);} |

## 3.6 GIS切图

全国航路图是一个非常大的PDF文档，在桌面系统上打开都需要等待一点时间，在Android平台上需要等待的时间更多，而且我们需要航路图的导航功能，所以我们需要把这个PDF转化成瓦片地图，结合经纬度信息，实现路径的勾画[[26]](#endnote-27)。

实现的硬件环境：

Machine1：

安装Windows操作系统，建议Win7 64位；

Machine2:

安装Linux操作系统，建议Ubuntu；

实现的软件环境：

* iServer 直接解压缩到Machine1；
* 安装GlobalMapper、SupermapDeskpro .net 6R、sqlite到Machine1；
* 安装FastDFS到Machine2；
* 将Machine1和Machine2连联通；

实现方式如图 15所示：



图 15 PDF转瓦片地图

实现步骤如下所示：

1.用GlobalMapper从PDF文档中获取图片，存储为a.ECW格式；

2.启动SupermapDeskpro .net 6R，导入a.ECW（默认为SDB数据集）;

3.创建UDB数据集，将a.ECW复制到UDB数据集，并将UDB数据集保存为b.smwu工作空间；

4.启动iserver；

5.通过IE访问iServer(htp://172.16.3.208:8090/iServer/)，将b.smwu工作空间发布到iServer；

6.通过IE访问iServer，设置切片功能；

7.启动切片功能；当图片较大或切割比例较小时，生产数据较大，需要等待较多时间；另外，切片后的碎片文件存储在FastDFS中，故Machine2需要留出足够的硬盘空间。

8.切片完毕后，可在IE界面上通过iServer将碎片文件下载到本地，存储为\*.Mbtiles文件（本质上是sqlite数据库）。

9.如果\*.Mbtiles文件在android程序中不可用，可通过调试信息logcat查找问题；（笔者遇到此问题时，发现Mbtiles数据库中的metadata表中的“crs\_wkt”栏读取异常，解决办法为：从可用的Mbtiles数据库中，将该栏数据拷贝并覆盖其内容，即可使用）[[27]](#endnote-28)。

需要注意的事项为：

1、Deskpro 及 iserver 获得许可（\*.lic文件）后，用 SuperMap License Manager 配置许可。

2、Deskpro操作（建议采用net.6R版本）：创建工作空间（smwu），导入数据（选择ecw文件），导入时，需要创建新的影像金字塔（约10分钟）。

3、iServer建议6R版本（高版本不支持 发布 Smwu格式工作空间）[[28]](#endnote-29)。

## 3.7 性能计算设计与实现

性能计算以起飞性能计算为例，飞机在某一个机场，比如首都机场，要起飞的时候，可以根据不同的跑道、飞机的空调是否打开、防冰是否打开、道面是干的还是湿的、襟翼是多少以及风速是多少来计算出温度、重量、速度以及限定条件等结果。以便驾驶员进行对照比较，根据比较结果，可能会在驾驶过程中，对某些方面进行特别的注意或者采取特别的措施，保证起飞的正常。

性能计算的过程是很复杂的，而且各个机场在地面的桌面系统上已经有比较准确完备的计算软件，已经将本机场的起飞性能的计算的条件和结果都保存在了数据库中。所以我们将采用将数据结果保存到sqlite数据库中，然后进行查询的方式来实现性能计算部分。

### 3.7.1 起飞性能数据库的获取

机场桌面系统中性能计算的结果保存在access数据库中，这也是我们能得到的数据，所以需要转化成Sqlite数据库。

首先使用DB2MYSQL软件来生成对应的SQL代码。DB2MYSQL是一个数据文件转化软件，它可以自动底地将一个ACCESS数据库文件转化为对应的SQL代码。它是一款可广泛应用于ACCESS数据库转换为MYSQL或其他类型的SQL数据库的软件。

DB2MYSQL能够实现可视化操作数据库，转换界面非常简单明了，可轻松实现ACCESS数据库转换为MYSQL数据库。如果想要将SQL SERVER的数据库转换到MYSQL，也可以使用这个软件，具体操作是先使用SQL SERVER的“导入和导出数据”将SQL SERVER数据导出到ACCESS文件，然后再将ACCESS文件导出为MYSQL语句[[[29]](#endnote-30)]。

运行转换功能后，系统会生成两个输出，一个是在屏幕界面上将转换后的SQL语句输出，另外还会在输出目录下自动生成一个后缀为.SQL的文本文件，里面会记录生成的SQL语句。DB2SQL使用的示意图如图 16所示。

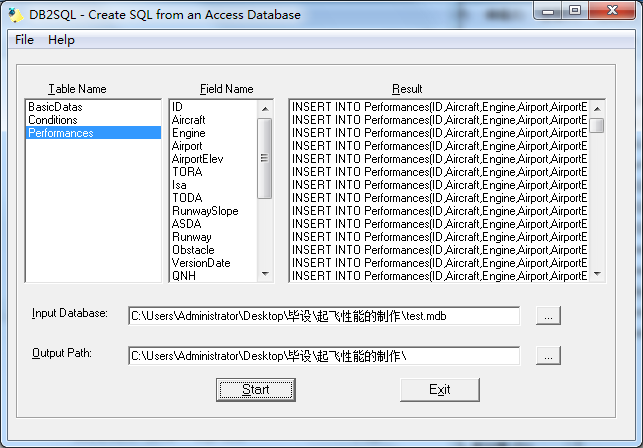


图 16 DB2SQL使用示意图

首先在桌面系统上，进入命令行，创建数据库，在数据库里创建数据库表。

数据库表的设计如表 6所示 ：

表 6 起飞性能数据库表设计

|  |  |  |
| --- | --- | --- |
| 表项 | 类型 | 备注 |
| ID | INT | 主键，自动递增 |
| Aircraft | char(13) | 机型 |
| Engine | char(18) | 发动机类型 |
| Airport | char(4) | 机场 |
| AirportElev | INT | 机场编码 |
| TORA | INT |  |
| Isa | INT |  |
| TODA | INT |  |
| RunwaySlope | char(6) |  |
| ASDA | INT |  |
| Runway | varchar(4) | 跑道（RW01等） |
| Obstacle | char(2) |  |
| VersionDate | char(18) |  |
| QNH | INT |  |
| AirCond | char(3) | 空调开关（ON、OFF） |
| AntiIcing | char(3) | 防冰（On/Off） |
| PCN | char(10) |  |
| Dry | char(3) | 道面（DRY，WET） |
| OATC | INT | 温度 |
| Weight | varchar(11) | 重量 |
| Limitation | char(3) | 限定条件 |
| velocity | varchar(9) | 速度 |
| CONF | varchar(3) | 襟翼 |
| WIND | INT | 风速（-10KT，0KT，10KT） |
| DataSource | char(4) | 数据来源 |

然后将数据导入数据库。

### 3.7.2 将数据导入数据库

导入数据的过程如图 17所示。



图 17 通过sql语句导入数据到sqlite数据库

## 3.8 导航数据库的设计与实现

导航数据库是地面支持系统中非常重要的一部分，它主要包括导航设备及报告点数据、机场及跑道数据、航路、公司航线和公共程序及公司特别程序（RNP、单发等）。导航设备主要类别有VOR、DME、NDB等，其地理位置以经纬度表示，标高以英尺为单位的海拔高度表示。其中VOR以三个英文字母表示，NDB、指点标以二个或一个英文字母表示，报告点以五字代码或P点表示。导航设备的级别分为：低高度级（L）、高高度级（H）、终端级（T）、无级别（UNC）。

机场地理位置以经纬度表示，标高是基于QNH，以英尺为单位的表示。跑道长度以英尺为单位表示，地理位置以跑道头坐标表示。航路分高空、低空和机场附近的终端航路等，航路数据包含航路类型、航路点说明（航路点坐标或作为航路点的导航台坐标等信息）。按照航空公司要求定制的航路，包括起飞机场、航路的走向、落地机场。每条公司航路包含以下基本信息：公司航路名称、沿航路的定位点、定位点类型及坐标、经由的航路。公共程序及公司特别程序包括标准仪表进场程序（STAR）、标准仪表离场程序（SID）、过渡和进近程序、精密仪表进近程序（ILS）、RNAV、RNP、单发。

目前导航数据库供应商特指持有FAA或JAA许可证的公司。持有一类许可证的公司有JEPPESEN、EAG、LIDO等。持有二类许可证的公司有JEPPESEN、HONERWELL、LIDO等。

下面将以上面提到的几个航空公司为例，来介绍导航数据库的制作流程。

HONEWELL导航数据库的制作流程为：JEPPESEN采集来自各国政府对外公布的各类航行情报资料（AIP、NOTAM等），JEPPESEN按ARINC424标准将数据制作成标准数据库提供给HONEYWEL，航空公司根据航班运行需要把公司航线、特别程序等数据提供给HONEYWELL，HONEYWELL把数据按其格式打包后提供给航空公司。

JEPPEEN导航数据库的制作流程为：JEPPESEN采集来自各国政府对外公布的各类航行情报资料（AIP、NOTAM等），航空公司根据航班运行需要把公司航线、特别程序等数据提供给JEPPESEN， JEPPESEN按ARINC424标准将数据制作成标准数据库提供给SMITH（GE），SMITH（GE）把数据按其格式打包后提供给JEPPESEN，JEPPESEN提供给航空公司。

导航数据库正常周期是28天更新一次，全年以供13个周期，数据库维护需要按照导航数据库公司提供的时间进度表执行。

基于Android平台的电子飞行包（EFB）系统的地面支持系统的导航数据库的制作的开发环境选用了C#和WPF作为开发的语言和开发的框架，后台数据库使用了Oracle数据库。

为了方便用户使用，本系统使用了现在微软比较流行的Ribbon和Docking Window。

在WPF中使用Ribbon使用的是WPF Ribbon Control控件库，开发出类似Office 样式的工具栏。微软专门发布了一款在WPF上的使用的Ribbon库称作Microsoft Ribbon for WPF，该Ribbon控件是完全属于WPF的工具。只需要在xaml文件中加入xmlns:ribbon="clr-namespace:Microsoft.Windows.Controls.Ribbon;assembly=RibbonControlsLibrary"，这个库空间就可以了。

Docking Window使用了AvalonDock控件。AvalonDock是一个专门为WPF设计的停泊窗口，它能够让你使用一个跟很多开发环境中使用的停泊系统一样的窗口创建自定义的布局，同时它还包括了主题和例子帮你快速上手。

程序的主要是流程是：从最开始，对数据文件（主要是xls文件和csv文件）进行核对检查，将机场的机场数据、跑道数据、离场数据、进场数据、进近数据、导航台数据、NDB台数据、航路点数据和等待数据导入后台Oracle数据库，当把所有资料都导入数据库之后，当有一个机场发来最新的数据文件的时候，就可以对这个文件中的数据跟后台数据中的数据进行对比，对比之后，修正后台数据库的数据。

在界面设计方面，由于导航数据库具有数据量大的特点，数据分析过程，以及对比过程如何将对比结果展示出来，让用户一目了然，同时便于修改，也比较的麻烦。

程序的主界面如图 18所示：

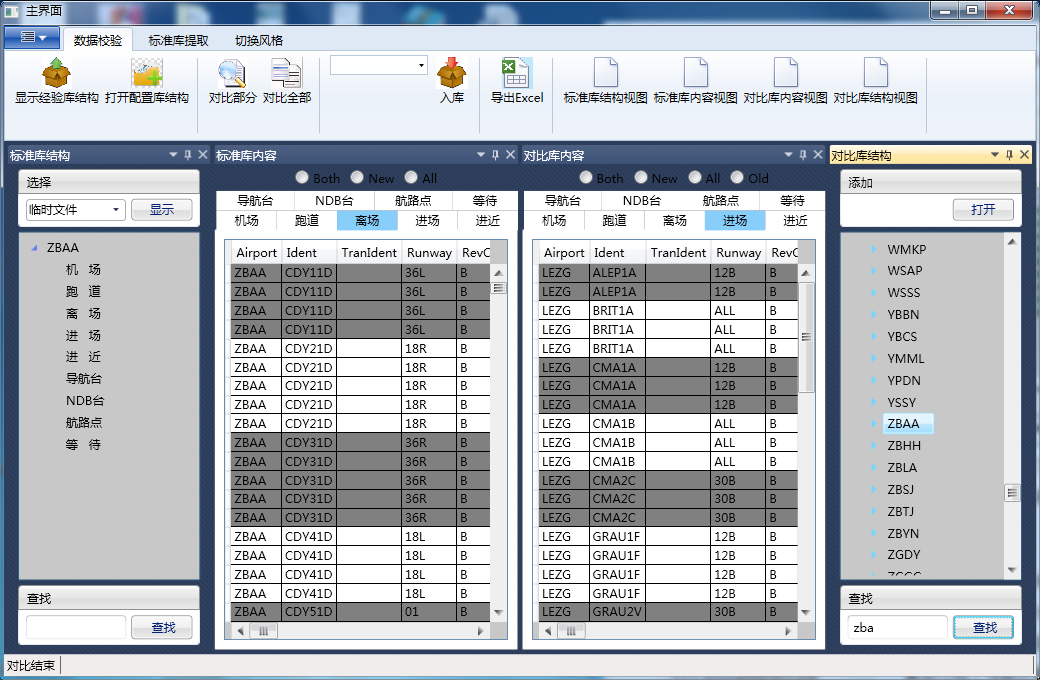


图 18 导航数据库软件界面

程序左侧打开了ZBAA也就是首都机场的数据，它是从标准库提取里面提取出来的，右侧是其他数据来源，然后进行数据对比之后，如图 19所示



图 19 导航数据库对比结果

结果中的左侧的颜色标识表示本机场中数据的一致性，其中B表示Both，N表示New，O表示Old，R表示Right。经过对比，会把某一个机场的某一项比如跑道数据的B，N，O，R表示出来，其中绿色表示完全一样，红色表示有差别。点击某一个机场的对比结果按钮，会将对比结果展示出来。

最后的对比结果如图 20所示。

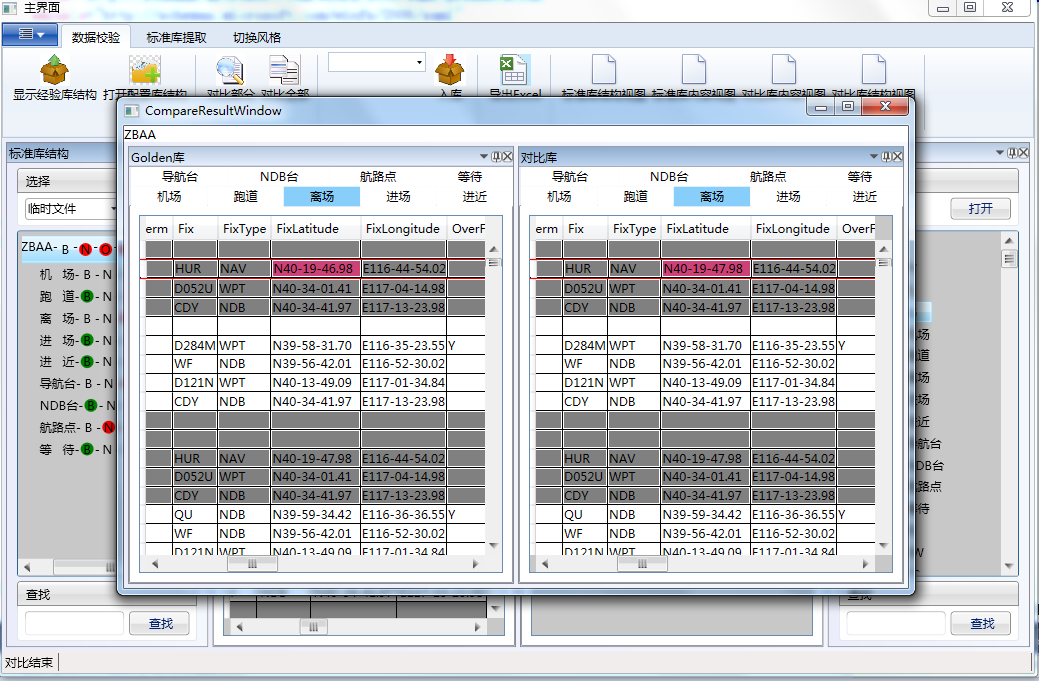


图 20 对比结果数据展示

程序设计过程中遇到了如下几个问题：

1. 内存溢出问题。

内存的溢出首先是因为数据本身是比较大的，每个机场的数据都读取到内存中就会占用100M到200M的内存空间。还有就是对比结果展示的时候，开始设计的时候，采取了渲染单元格的方式，这样的话，就导致了将所有的数据都渲染出来，这样内存占用很快就会达到2G甚至更多。

1. 对比结果展示的问题。

对比结果中那些不同项需要用不同的颜色展示，这是很容易做得到，但是离场、进场和进近的原始数据本身就是Runway的值来区分颜色的，而且如果某一行中如果有不同的项，由于每一行特别的长，如果用户每查一行都需要来回拖动数据，也会比较麻烦，所以需要让有不同数据的行本身用不同颜色的框包括起来。这样，多种需求结合在一起，会使得程序设计比较麻烦。

面对数据量大，显示需要清晰的问题，在设计导航数据库的时候，主要采取了如下设计方案。

1. 数据的磁盘存取。

这个主要是应对机场数据比较多的问题设计的。为了避免多个机场的数据一起读取进入内存，导致内存被消耗光的问题，我们只让一个机场数据在内存中，当切换机场的时候，只需要把当前机场的数据以一定的格式保存到磁盘上，设计的格式，可以确保当再次读取这个机场的数据的时候，可以快速的读取进入机场。

1. 数据绑定。

WPF中的数据绑定提供了很强大的功能。与普通的WinForm程序相比，其绑定功能为我们提供了很多便利，例如Binding对象的自动通知/刷新，Converter，Validation Rules，Two Way Binding等功能，省去了很多维护的繁琐工作。另外对于WPF中提供的数据模板功能，让我们可以轻松定制可以被复用的控制呈现的模块—但这是以数据绑定为前提来做到轻松易用的效果的。数据提供者例如XmlDataProvider和ObjectDataProvider更是简化了将对象以特定方式绑定并呈现的过程。可以说，数据绑定是WPF中让我们真正能够开始体现其便利性的特征之一，而对以数据驱动的应用来讲，其重要性不言而喻。通过将数据结果动态绑定让他们自动刷新，这样的话，对比结果就不需要全部读取进内存，也就不用使用渲染单元格的方式来进行结果的展示。让结果自动刷新，既使得内存使用率小了，也使得代码可维护性增强[[[30]](#endnote-31)]。简单来讲，数据绑定是在应用程序UI与业务逻辑之间建立连接的过程[[[31]](#endnote-32)]。

数据绑定实质上是绑定目标与绑定源之间的桥梁，每个绑定都具有四个组件：绑定目标对象、目标属性、绑定源，以及要使用的绑定源中的值的路径目标属性必须为依赖项属性，绑定源对象并不限于自定义CLR对象。绑定模型如图 21所示

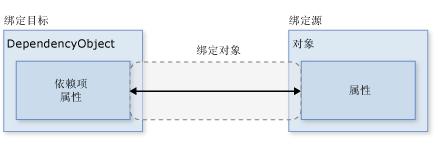


图 21 WPF数据绑定模型模型

绑定的模式有OneWay、TwoWay和OneWayToSource等。[OneWay](http://msdn.microsoft.com/zh-cn/library/system.windows.data.bindingmode(v=vs.110).aspx) 绑定导致对源属性的更改会自动更新目标属性，但是对目标属性的更改不会传播回源属性。[TwoWay](http://msdn.microsoft.com/zh-cn/library/system.windows.data.bindingmode(v=vs.110).aspx) 绑定导致对源属性的更改会自动更新目标属性，而对目标属性的更改也会自动更新源属性。[OneWayToSource](http://msdn.microsoft.com/zh-cn/library/system.windows.data.bindingmode(v=vs.110).aspx) 与 [OneWay](http://msdn.microsoft.com/zh-cn/library/system.windows.data.bindingmode(v=vs.110).aspx) 绑定相反；它在目标属性更改时更新源属性21。绑定模式如图 22所示。

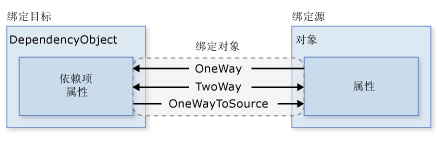


图 22 WPF数据绑定模式

## 3.9 本章小结

本章先介绍了电子飞行包系统的总体设计，然后分别介绍了电子飞行包系统中各个功能模块的具体设计路线，主要是阅读器的设计与实现、数据增量更新的设计与实现、机场飞行滑跑导航的设计与实现、飞行计划导航的设计与实现、检查单的设计与实现，最后介绍了性能计算设计与实现和地面支持系统中的导航数据库的设计。

# 第四章 电子飞行包功能性测试

## 4.1 测试环境部署

本节介绍基于Android平台的电子飞行包测试环境，包括测试数据的介绍以及测试环境介绍。接着，本章将按照功能模块来测试电子飞行包（EFB）系统的稳定性。基于Android平台的电子飞行包主界面如图 23所示。



图 23 Android平台的电子飞行包主页面

### 4.1.1 测试数据介绍

电子飞行包系统主要是实现纸质资料的电子化，所以会需要很多的文档，同时本文在设计基于Android平台的电子飞行包的时候加入了导航等功能，所以本地存储的资料特别多，种类也特别多。

1. 机场航图资料。

航图资料有10多种，分别给每一种资料编号，在每个机场资料的文件夹下，有一个index.txt的文件。它的每一行的数据格式为：

PDF文件名 编号 航图的名字

这样，当程序中查阅某一个机场的资料的时候，就可以根据这个index.txt文件将资料进行进行快速的定位展示。目录结构如图 24所示



图 24 电子飞行包航图资料目录安排

1. 机场图

每一个飞机场都有一个机场图，它们是smwu格式的文件，同时还有udb和udd两个数据集。它们也是放在各自的机场四字代码名的文件夹下。

1. 公司资料

主要是一些规章标准和机型手册资料。它们将单独放在一个文件夹下。

1. 飞行计划

这里面存储着飞行计划的电子版和数据库版。

1. 性能数据库

性能计算所需要的数据库都放在这里面。

### 4.1.2 测试环境介绍

测试环境在Android端就是基于Android平台电子飞行包应用程序。在地面支持系统中，选用32位win7系统作为服务器环境，使用了Visual SVN Server和TortoiseSVN机型数据的操作。

## 4.2 测试过程及结果

### 4.2.1 文档查阅测试

文档的查询可以在资料模块，输入要查询的机场的四字代码、机场名字或者三字代码来查询，如图 25所示。



图 25 机场航图资料查询

对每一个机场，按照航图资料的类别分别把他们放到不同的类别下面，并且按照编码顺序排放，同时软件中显示的是资料用途的名字，而不是资料的文件名，这样就会一目了然，更加方便用户的资料查询。

此外在起飞机场模块和进场部分模块，只是对此次航行的起飞机场和降落机场的资料进行分类整理，如图 26所示。

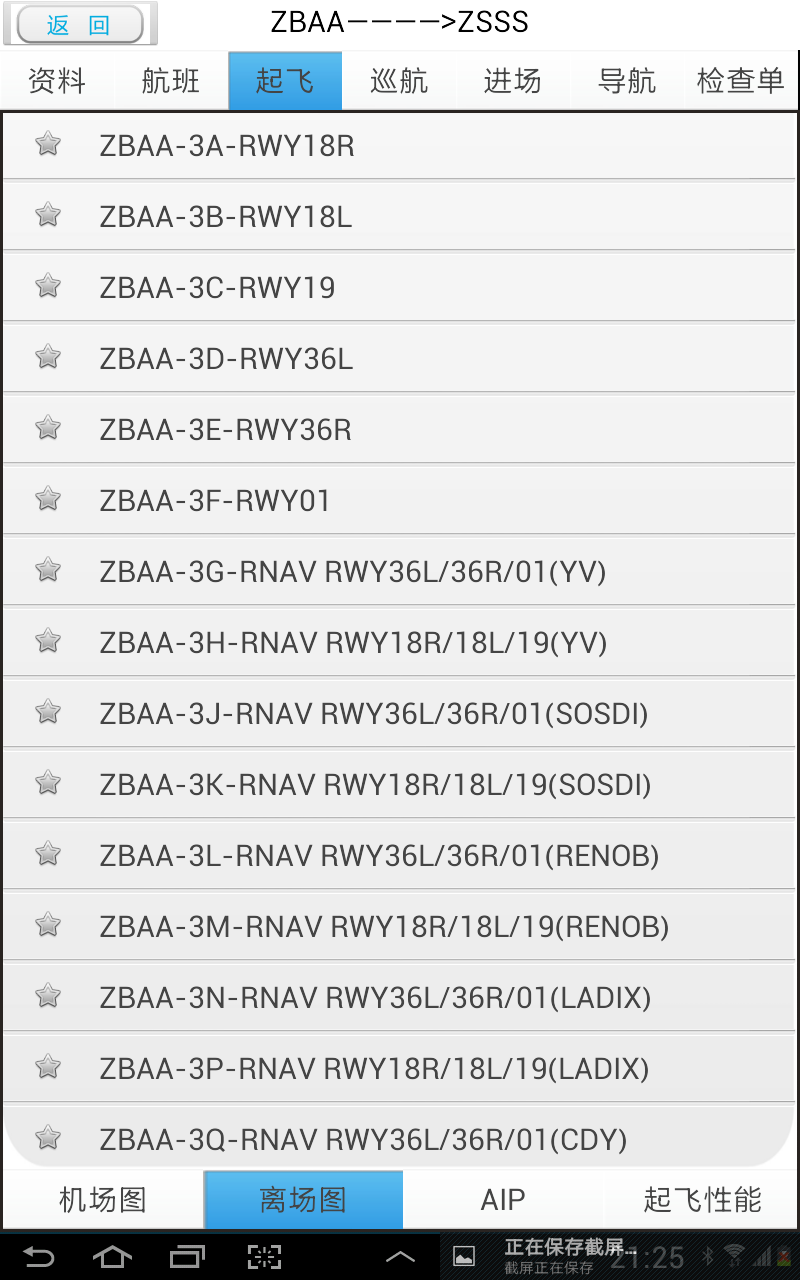


图 26 起飞机场资料分类展示

这两个模块主要是方便用户对其飞机场资料和降落机场资料的快速查询，同时还包括了起飞性能和降落性能的设计和实现。

### 4.2.2 起飞性能计算测试

起飞性能计算主要是通过一些系列的参数，计算出速度、限定条件、温度或者重量。如图 27所示。



图 27 起飞性能计算

### 4.2.3 切图测试

全国航路图是一由一个非常大的PDF转化而来，最终将一个PDF图转化成了瓦片地图，保存在了mbtiles里面，由于使用的瓦片地图，地图打开比较快，缩放也比较快。全国航路图展示，如图 28所示。

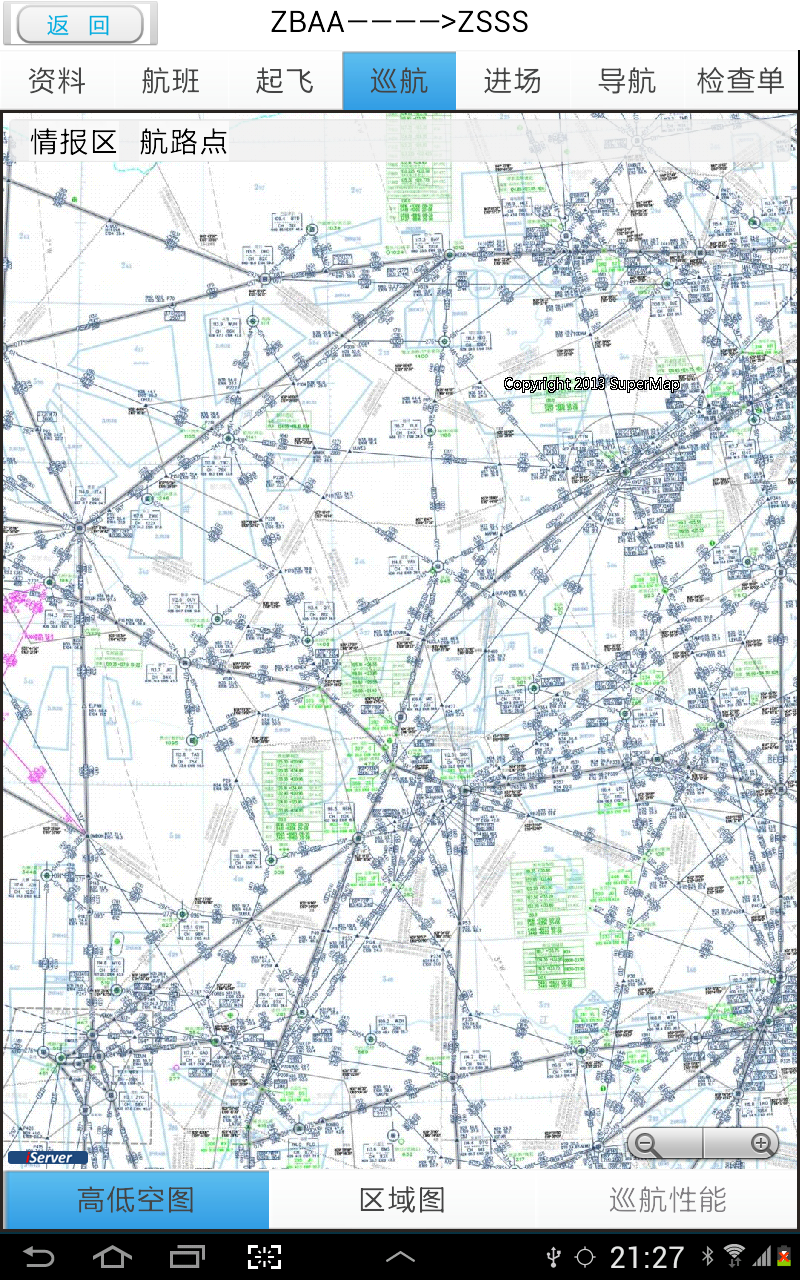


图 28 全国航路图展示

### 4.2.4 飞机滑跑测试

飞机在机场滑跑的时候，会根据命令沿着指定的路线进行滑跑，同时根据GPS的定位信息知道自己的位置所在，根据GPS定位信息可以看出飞机与指定的跑到之间的距离，从而判断飞机是否正在沿着正确的跑到滑行。如图 29所示。

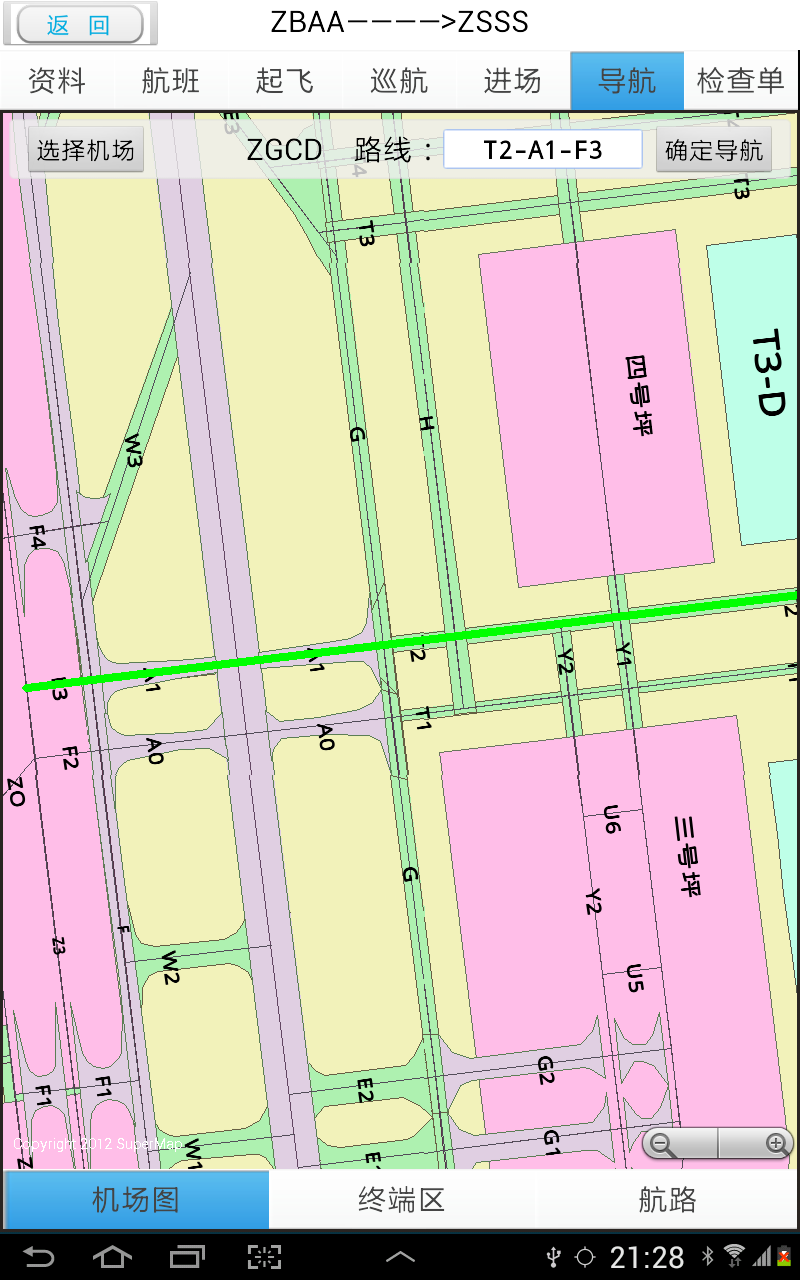


图 29 飞机机场滑跑

### 4.2.5 飞行计划导航测试

飞行计划导航部分，以从北京飞往九寨为例，根据飞行计划中的信息，将从北京到九寨之间的所有的航路点和情报区画了出来，点击GODON航路点，在屏幕的左上角出现了该航路点的具体信息，主要有TMP（温度）、FUEL（油耗）和WIND(风速)等。如图 30所示。

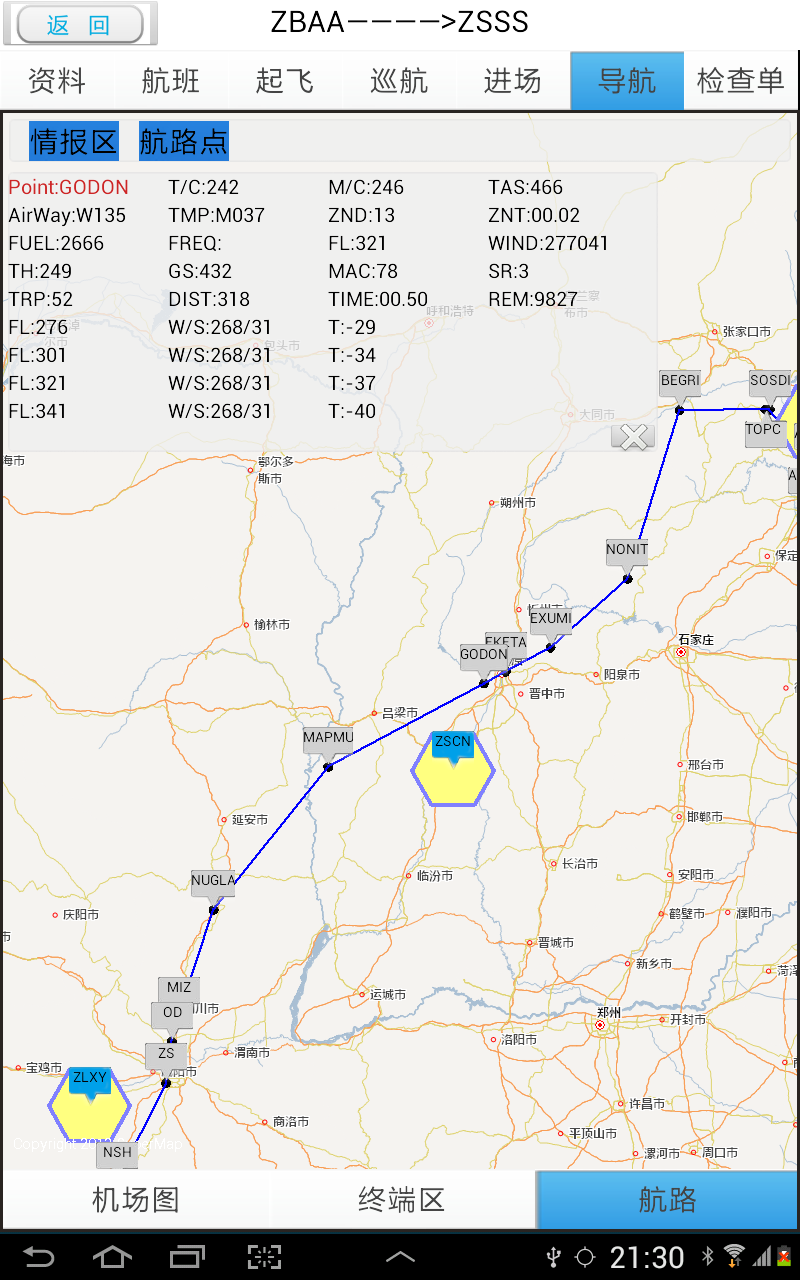


图 30 飞行计划导航示意图

### 4.2.6 检查单测试

检查单主要是对飞机飞行的各个阶段的过程中，对飞机的一些检查。比如在起动前，就会有驾驶舱准备、起落架销子和堵盖、旅客信号牌和燃油量等的检查。如图 31所示。



图 31 检查单使用示意图

使用过程中会发现，只有检查了第一条，也就是驾驶舱准备之后，才可以去检查起落架销子和堵盖，也就是说，检查的顺序必须是按照顺序来的，不会出现乱序检查。在检查完起飞前后，点击起飞后的时候，会发现下方的检查项会自动的滚动，减少了操作。

## 4.4 本章小结

本章对基于Android平台的电子飞行包（EFB）系统进行了详细的功能测试。主要包括文档查阅测试、起飞性能计算测试、切图测试、飞机滑跑测试、飞行计划导航测试和检查单检查测试。通过测试，程序运行流畅无误，相应比较快，功能准确，用户使用方便友好。

# 总结与展望

工作与研究总结

当前，电子飞行包是航空业界的一个研究热点，目前的电子飞行包系统大部分都是基于IOS平台开发的，随着Android系统的迅猛发展，越来越的第三方功能模块和控件可以使用，大大方便了Android开发者进行应用程序的开发。同时还有Android设备相比IOS设备会便宜许多，可以选择的设备类型也比较多，所以开发基于Android平台的电子飞行包是目前航空业界开发商的选择。

本文调研分析了国内外很多公司比如JEPPESEN公司的电子飞行包的设计，分了各个公司电子飞行包设计的使用流程、功能和使用平台，深入研究了开发基于Android平台的电子飞行包的设计与实现的可行性，对基于Android平台的电子飞行包的各个需要的模块的将要选用的技术进行了对比分析，通过对各个技术模块的对比分析，确定了模块实现的可行性和最终的模块实现方式。并针对目前大部分电子飞行包中缺少的飞行计划导航问题，提出了飞行计划信息与SuperMap GIS结合的方式，以图形的直观的方式对驾驶员进行飞行导航，最终设计实现了基于Android平台的能够进行资料电子化、资料的增量更新、飞行信息数据与GIS结合导航、电子化检查单检查、机场滑跑定位导航和数据资料快速查阅的电子飞行包，同时还提供了导航数据库系统作为地面支持系统。

本文具体的研究成果包括：

1. 分了电子飞行包的含义，对电子飞行包的硬件和软件分类进行了阐述，分析了电子飞行包中那些需要进行电子化的资料进行了分析，设计出了合理的资料目录，对各种GIS图合理的目录结构组织。
2. 对电子飞行包系统中各个功能模块的技术实现，进行了对比分析，来确定最终的技术实现。主要是对Android平台上的各种开源PDF阅读器进行了对比分析、对数据增量更新解决方案进行了对比分析、对Android系统的应用程序内存分配进行了分析、对SuperMap GIS进行了分析并与飞机在机场滑跑和飞行计划导航进行了结合，最后对地面支持系统的开发框架主要是MFC和WPF进行了对比分析。
3. 最后分析了基于Android平台的电子飞行包的使用流程和各个功能模块的使用方式，并总结了使用基于Android平台的电子飞行包的各个功能的方式。

对未来工作的展望

首先，安装了Android系统的只能及占有的市场份额越来越多，Androdi设备的增加，使得Android开发者的数量和Android平台上的应用程序的数量都大大增加，有了广大开发者的支持和广大设备厂商的支持，在Android平台上开发应用程序，会有更多可选的控件或者库，它能够极大提高程序的开发效率，提高程序的安全性。

其次，电子飞行包系统的开发已经是电子化航空公司的必备条件，基于现在IOS设备比较贵以及现在基于Android平台的电子飞行包（EFB）系统匮乏的现状，开发基于Android平台的电子飞行包系统是非常有前景的。电子飞行包系统可以改变人们传统的工作方式，将很多工作都设计到电子飞行包系统中。电子飞行包系统离不开地面支持系统，地面支持系统的创建和完善也是不可或缺的一部分，所以以后会丰富电子飞行包系统的功能，同时会不断晚上地面支持系统，开发出很多具有自主知识产权的产品。

# 参考文献

# 攻读硕士学位期间取得的学术成果

北京航空航天大学. 基于Android系统的EFB导航系统及使用方法[P]. 姜博,盖玉杰,龙翔,高小鹏等. 中国专利.已受理. 专利申请号：21410698292.0. 2014-11-28

# 致 谢

在本文即将完成之际，我的研究生生活也即将结束。回顾研究生两年多的生活，心中感慨万分，收获颇多。在这难忘的这两年时间里，我增长了知识，收获了珍贵的友谊，自己得到了很大的锻炼和成长，为此，我要感谢各位老师、同学和家人的关心和帮助。

首先要感谢我的导师龙翔教授。龙老师渊博的知识、严谨的治学态度、正直的作风给我们留下了深刻的印象，为我们丰富了学识，开阔了视野，鞭策着学生不断进取、前进。

另外我还要感谢姜博老师的悉心指导和帮助。王老师渊博的知识、丰富的经验，令人尊敬的品德、待人处事的谦和态度，使我终身受益。您将是我今后学习、生活和工作的榜样。

此外，在论文及答辩的撰写过程中也得到了姜博老师的指导和帮助，姜博老师给本文及答辩演示等提出了准确、详尽、有益的修改意见，姜老师认真负责的态度，值得我们学习。另外，感谢万寒老师指导我的工作。

感谢项目组的吴兴博师兄。在我整个研究生阶段，吴兴博师兄给予了我各个方面的照顾和帮助。在学习方面，吴兴博师兄在开题、中期、完成毕业论文的各阶段都给予了指导和帮助。在我找工作的过程中，师兄也给与了很多建议。

感谢项目组李力，于雷，跟他们一块学习，我收获很大。各个同学的知识面各不一致，让我了解到了计算机其它方向的有趣的知识。感谢李力给予我的帮助，在GIS系统使用分布式切图的模块，李力同学帮助我进行测试发开。感谢于雷同学对飞机机场滑跑路线设计方面的思路。

同时要感谢实验室同学孙剑文、徐晓丹、张弛、陈鹏、阮帅、马春雷、郭兰等，实验室学术气氛活跃，每个人的学习能力、科研水平、职业素质都在提高。同你们一起的日子里，我学到了很多东西。和大家一起吃饭、一起打羽毛球、一起讨论问题的美好时光，这也是我人生的一笔财富。

感谢在网络上一些热心的技术大牛。他们在工作之余，能够分享自己积累的技术知识，热心回答技术问题，他们经验丰富而又态度谦和，是我学习的榜样。

我要特别感谢我的父母，感谢父母的养育之恩，感谢他们无微不至的关心和照顾，他们的不断鼓励和忘我付出是我不断前进的动力，他们始终是我坚强的后盾。

最后，感谢各位评审老师在百忙中抽出宝贵时间对我的论文进行评阅和审议。

1. [] 杨阳，张诗翱 浅析电子飞行包 《科学资讯》2014 NO.24 [↑](#endnote-ref-2)
2. [] 吕小平 电子飞行包(EFB)系统介绍 《中国民用航空》 2007-10-15 第一页 [↑](#endnote-ref-3)
3. [] 王鑫 移动互联网的现在和未来 《中国邮政报》 2014-04-01 [↑](#endnote-ref-4)
4. [] Android内存研究 <http://www.cnblogs.com/wanqieddy/archive/2012/07/18/2597471.html> [↑](#endnote-ref-5)
5. [] APV pdfviewer主页 http://code.google.com/p/apdfviewer/ [↑](#endnote-ref-6)
6. [] 开源Androidpdf开发总结

   <http://www.cnblogs.com/pokeGame/archive/2011/06/02/2068575.html> [↑](#endnote-ref-7)
7. [] 不详. Palm被诉侵权muPDF阅读器是关键J. 中国信息化, 2009, (24):13-13. [↑](#endnote-ref-8)
8. [] MuPDF主页 <http://www.mupdf.com/> [↑](#endnote-ref-9)
9. [] 戴楠, 闫明星. 用SVN实现软件的版本控制J. 电脑知识与技术, 2009, (16). [↑](#endnote-ref-10)
10. [] 胡晓锋. 基于SVN的网络协同开发环境构建J. 民营科技, 2010, (6). [↑](#endnote-ref-11)
11. [] 易文龙, 华晶, 何火娇. SVN版本控制在软件工程专业实训的应用J. 农业网络信息, 2010, (9):39-41. [↑](#endnote-ref-12)
12. [] GIS系统-地理百科 http://wiki.chinageo.com/doc-view-64792.htm [↑](#endnote-ref-13)
13. [] OsmAnd Google Code https://code.google.com/p/osmand/ [↑](#endnote-ref-14)
14. [] OSMAND展示 http://wiki.openstreetmap.org/wiki/OsmAnd [↑](#endnote-ref-15)
15. [] OSMDroid Github https://github.com/osmdroid/osmdroid [↑](#endnote-ref-16)
16. [] OSMAND图示 https://github.com/osmdroid/osmdroid [↑](#endnote-ref-17)
17. [] Esri China http://www.esrichina-bj.cn/softwareproduct/ArcGIS/ [↑](#endnote-ref-18)
18. [] ArcGIS 百度百科 http://baike.baidu.com/item/ArcGIS?fr=aladdin [↑](#endnote-ref-19)
19. [] Chun-hua H, Fu-ping L. Tailing Resources Management System Based on SuperMap GISJ. Onrvaon and Lzaon of Mnral Ror, 2011. [↑](#endnote-ref-20)
20. [] Chun-hua H, Fu-ping L. Tailing Resources Management System Based on SuperMap GISJ. Onrvaon and Lzaon of Mnral Ror, 2011. [↑](#endnote-ref-21)
21. [] 夏凉，张瑞 《MFC与WPF之比较》 China science and Technology Review 2012 [↑](#endnote-ref-22)
22. [] 江一火，李云华 《基于WPF的在线笔记本系统的研究与实现》 《宁德师范学院学报（自然科学版）》-2013年3期 [↑](#endnote-ref-23)
23. [] MuPDF多线程http://www.mupdf.com/docs/overview [↑](#endnote-ref-24)
24. [] VisualSVN Server http://wenku.baidu.com/view/8ca72deb19e8b8f67c1cb927.html [↑](#endnote-ref-25)
25. [] TortoiseSVN中文使用手册http://blog.csdn.net/xilidecai/article/details/4566660 [↑](#endnote-ref-26)
26. [] 王桥, 罗海江, 李茜. 超图软件——基于Supermap GIS的中国环境监测空间数据库J. 计算机光盘软件与应用, 2010, (20):58-62. [↑](#endnote-ref-27)
27. [] 刘冰, 谢轲, 陈小乐等. 基于GIS的瓦片式地图切图算法的设计与实现J. 科技信息, 2011, (7). [↑](#endnote-ref-28)
28. [] 王小军, 刘璐. 基于ArcGIS Engine进行瓦片式切图的技术研究J. 测绘与空间地理信息, 2010, (4). [↑](#endnote-ref-29)
29. [] Access数据库转MYSQL http://blog.sina.com.cn/s/blog\_6901eb650100yruz.html [↑](#endnote-ref-30)
30. [] WPF数据绑定 http://www.cnblogs.com/zlgcool/archive/2008/10/22/1316605.html [↑](#endnote-ref-31)
31. []<http://msdn.microsoft.com/zh-cn/library/ms752347(v=vs.110).aspx#basic_data_binding_concepts> 数据绑定概述 [↑](#endnote-ref-32)