

# 移动环境下面向电子地图服务的客户端系统设计与实现

## 摘 要

21 世纪是不同领域科技创造性集成和融合的时代。随着地图制图技术、地理信息系统 (GIS) 技术与主流 IT 技术的融合, 极大的推动了电子地图服务的发展。电子地图服务已经越来越广泛地被人们所利用, 在交通、导航、旅游等各个领域发挥着重要的作用。近年来, 随着移动通讯技术的飞速发展, 移动环境下的电子地图服务应运而生, 为人们提供着日益广泛的在线、移动的地理信息服务。

然而, 目前大部分的电子地图服务只是提供诸如一般的导航、查询等公共的地图服务, 这些服务往往很少与用户的自身情况相联系。因此, 如何根据移动用户自身属性与特点为用户提供个性化的电子地图服务成为该领域的一个重要研究问题。基于上述问题, 本论文分析了移动环境下的电子地图服务的特点, 在移动终端上设计并实现了基于用户属性的定制化的电子地图服务系统。

论文首先介绍了移动终端电子地图服务客户端系统的研究基础和功能需求。在需求分析的基础上详细描述移动环境下电子地图服务客户端系统整体结构, 然后详细描述电子地图中具有代表性的服务——用户兴趣点 (POI) 服务的开发过程, 客户端系统采用诺基亚 N800 移动终端, 将开发后的程序安装部署到 NOKIA N800 上, 通过 N800 互联网终端呈现移动环境下的电子地图服务。

最后对客户端系统 POI 服务的功能与效率进行测试, 并对现有客户端系统实现的电子地图服务的不足提出改进方案。

**关键词:** 电子地图, 地理信息系统 (GIS), 兴趣点 (POI), Maemo, Hildon

# DESIGN AND IMPLEMENTATION OF E-MAP SERVICE ORIENTED CLIENT SYSTEM IN MOBILE ENVIRONMENT

## ABSTRACT

Nowadays it becomes more and more evident that science and technology from different fields are being creatively integrated. Due to the integration of map graphic technology, GIS and IT technologies, electronic map (E-map) oriented services come out and provide an increasingly wide range of online mobile geographic information services to the public. With the development and widely use of communication technologies, E-map oriented services have been greatly promoted.

However, current E-map services only provide public map service such as searching and guiding, and this service has little relation with the personal information of a target user. Therefore, how to provide a customized map services according to the information of a mobile user becomes an important work. Based on this consideration, the thesis analyzes the features of E-map services in mobile environment, and designs as well as implements the client system and an example E-map service on the mobile terminal of Nokia N800.

The thesis first introduces the background knowledge of the E-map services and the client system. Then the general architecture of the E-map services and the requirements for implementing the E-map services in the client system are analyzed. Following this, the detail design and implementation of the client system and an example E-map service are emphasized. Finally, some test results are given.

In addition, several optimization schemes are also provided in the thesis according to the analysis of the current system.

## KEY WORDS

Electronic Map, GIS, POI (Point Of Interest), Maemo, Hildon

## 声 明

### 独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：  日期： 2009.3.12

### 关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在\_\_年解密后适用本授权书。非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：  日期： 2009.3.12

导师签名：  日期： 2009.3.13

# 第一章 绪论

## 1.1 论文背景

近年来，移动通信技术不断蓬勃发展，3G、4G 概念相继提出，各种移动终端也在不断升级和普及，移动信息服务逐步深入到人们生活的各个角落，这给数字地图和地理信息系统（GIS）在面向移动终端上的应用又提出了新的需求。随着电子技术、网络技术、通信技术、计算机技术、空间信息技术的发展，为真正的随时随地的移动信息服务提供了理论和技术支持。

电子地图应用是利用成熟的网络技术、通信技术、GIS 技术，实现地图服务的方式[1]。它综合运用全球定位系统（GPS）的精确定位技术、便携移动设备、无线接入和 GIS 的空间信息处理能力，使得人们能够实时地获取、存储、更新、处理、分析和显示地理信息。移动环境下的电子地图服务作为移动地理信息系统中的重要应用之一，具有位置配准、信息查询、路径生成等作用，在公交、导航、旅游等领域应用十分广泛。因此，移动环境下的电子地图服务也成为了当前学术界研究的热点。

目前大部分的电子地图服务只是提供单纯的地图查询、兴趣点搜索等服务，这些服务往往很少与用户的自身情况相联系。而本论文介绍的移动环境下的面向电子地图服务充分利用了移动终端的特性，不仅与用户的具体情况相联系，还在用户与用户、用户与服务器之间产生相应的互动，从而实现了更加新颖并且实用的电子地图服务。

## 1.2 论文工作

移动环境下面向电子地图的服务系统主要分为客户端系统与服务器系统两大部分。本论文的主要内容是针对客户端系统的开发与实现，即在移动终端上设计和实现可交互性的电子地图服务系统，包括客户端系统的整体结构设计，端系统模块设计与划分，用户界面布局的设计与开发、部分功能模块的详细设计与开发编码、端系统数据库系统设计和通信接口的设计与实现。

### 1.3 论文的组织

论文共分七章，组织如下：

第一章，绪论。简要介绍了项目的主要理论背景、论文的主要内容和论文的组织。

第二章，介绍提供电子地图服务的系统概况。描述了电子地图基本服务类别，介绍了电子地图系统相关概念与原理。包括地图分块缓存技术、图像生成技术、坐标投影技术等。并且描述了客户端与服务器端系统组成，以及服务器端的组件类型与业务流程。

第三章，介绍移动环境下面向电子地图服务的系统整体结构模型，以及服务器与客户端系统组成与简要说明。

第四章，详细介绍客户端系统的需求分析和总体设计。描述了客户端系统的功能需求和接口需求等。并根据需求分析说明客户端系统的设计目标和原则，着重阐述了客户端系统的逻辑结构、总体设计、功能模块划分和功能描述。

第五章，介绍客户端系统中用户兴趣点（POI）功能模块的详细设计和实现。阐述了该模块的详细设计思想、模块的详细逻辑结构和主要数据结构、关键技术以及部分实现细节等内容。

第六章，面向移动电子地图客户端系统测试，包括主界面呈现、功能展示、典型测试用例以及测试结果等，然后对分析了客户端系统存在的性能问题，以及可以进行功能及性能改进的地方。

第七章，总结和展望。对论文工作进行了总结，并对下一步的工作提出可行性的建议。

## 第二章 电子地图系统概述

电子地图系统是指在计算机软硬件的支持下,以地图数据库为基础,能够进行空间信息的采集、存储、管理、分析和显示的计算机系统。它是传统地图与计算机技术、GIS 技术、网络技术相融合的产物,是地理信息的符号化表现,是空间信息与专题信息的结合,是提供地理信息公共服务的重要渠道[2],具有更丰富的信息含量和广阔的应用范围。

### 2.1 电子地图系统包含的主要服务

电子地图利用成熟的网络技术、通信技术、GIS 技术,实现了一种新的地图服务方式,具有位置配准、信息查询、路径生成等作用,包括数据采集服务、位置服务、导航定位服务等几个方面。

#### 1、数据采集服务:

数据采集服务为交通、运输、导航、公用设施管理、环境监测、公共卫生、应急响应、灾害处理等动态应用系统提供野外信息采集、数据更新和 GIS 应用平台;为基础设施制图(如学校位置、道路网络、森林边界)提供一体化的数据采集、编辑、成图手段;为逐家挨户的货物派送、信件邮递、健康调查、社区统计、社会经济数据采集和建库提供便捷的手持工具。

#### 2、位置服务:

位置服务为基于位置的服务(location-based services),它提供个性化(Personality)、本地化(Local)、实时(Real-time)的地理信息服务。其原理是利用手机(蜂窝式基站)或 GPS 设备定位,将位置信息发送给服务器,在 GIS 服务器完成数据库查询,返回给终端用户邻近的兴趣点的信息(旅馆、加油站、市场、停车场等)。

#### 3、导航定位服务:

导航定位服务一般由 GPS 进行定位。GPS 定位又分为车辆跟踪系统(Vehicle tracking system)和自主导航系统(Self-navigation system)[3]。车辆跟踪系统由 GPS 接收机、无线通讯设备(GSM)组成,将 GPS 位置信息通过无线通讯网络传给远程控制中心,并在控制中心的 eMap 上显示出其位置。自主导航系统由 GPS、内置的 eMap、地图匹配软件(如 Garmin 的 Mapsource、Lowrance 的 Mapcreate、Thales

Navigation 的 Mapsend manager)、无线通讯设备(GSM)组成, 移动用户不仅可以利用 eMap 进行位置配准, 还可以在 eMap 上进行位置查询、行程回顾、路线设计和兴趣点查询, 提供兴趣点和方向变化的语音提示。

## 2.2 电子地图系统相关概念与原理

电子地图系统开发相关技术与方法包括地图缩放级、地图分块与缓存技术、地图预先生成技术、坐标投影等, 这些技术在电子地图系统中发挥着重要作用。

### 1、缩放级

电子地图有多个离散的地图缩放级别。从 0 级到 17 级, 按照从大到小的顺序, 逐渐提供由概略到详细的地图服务。对于每个缩放等级, 都预先定义好相应的比例尺与之对应。正是这些预先定义好的离散的缩放级别为地图的分块提供了可能性。

### 2、地图分块与缓存

地图分块技术是将地图容器分割成  $256 \times 256$  像素大小的若干正方形地图方块, 每个地图方块都处在该地图容器的地理环境中, 并拥有一定的具体参数: 如缩放级别、投影类型和地理坐标, 通过算法由处理引擎根据这些参数分别向服务器请求地图图片来填充, 并且负责这些地图方块的无缝拼接、整体移动和地图填充。当用户做出一定的动作时(如平移、放大或缩小), 地图处理引擎根据一定的算法计算出需要新加载的小块地图, 并异步多线程的向服务器发出请求。最后, 当地图贴片传回用户端时, 再由处理引擎无刷新的无缝拼接成用户界面中的大地图, 下次使用时则不用向服务器再次请求, 直接利用缓存中的图片即可。

以地图漫游为例, 用户可以用鼠标拖拽地图来进行地图漫游, 处理引擎控制各个地图方块作为整体随着鼠标拖拽方向而移动, 如果移动得足够远, 要显示一些新的区域时, 这些区域的小块地图图片将会异步加载。在地图的更新过程中, 用户可以继续移动地图进行漫游, 触发更多的更新。这些小块地图在用户的会话过程中会被缓存, 从而使得当回到以前曾经访问过的地图时, 显示速度非常快。

分块的大小并没有一般性的标准可循, 可以按比例任意规则分块, 但在应用中则不得不考虑某些实际问题, 比如方便程度、效率、磁盘读/写等。一幅图像分块的多少对显示速度有直接的影响, 块数一般根据具体情况和多次试验取经验值, 假设分块大小为  $256 \times 256$  像素, 一般用户屏幕分辨率为  $1024 \times 768$ , 则显

示 1 屏地图最多需要 12 块。

### 3、地图预先生成

电子地图服务很多时候对地图的实时性要求并不高，道路交通、行政边界、河流等的变化就比较慢，定期如 1 个月更新一次地图就能满足需求。而且地图上的一些实时信息也不是固化在地图上的，而是通过异步读取数据库中的信息并通过简单的图层叠加实现的。而基础数据（高级别的道路、桥梁、城市、湖泊等）则是很少变动的，即便是某些道路发生变化，也可以很方便的生成该范围的地图斑块，这个工作量是比较小的。

因此，对用户访问量大、实时性要求不高的普通地图、专题地图和卫星影像地图可以预先生成。目前比较通用的做法是：对每个缩放级别，在约定的地理坐标和投影类型下，把地图切割成尺寸为  $256 \times 256$  像素的若干小块栅格地图贴片，并保存在地图贴片服务器中，并且通过算法将客户端地图方块坐标和贴片文件位置及名称关联起来。地图贴片格式可以多种多样，如 PNG、GIF、JPEG 等。

### 4、像素空间

为了将地图的地理坐标和客户端屏幕的像素联系起来，可以将地理坐标（经度/纬度）映射到一个假象的像素空间上。在计算机屏幕上，是通过像素来显示图像，地图贴片作为栅格图像也是用像素来表示其尺寸，因此使用像素空间可以很容易的排列地图贴片。

### 5、坐标和投影

一个大地测量学的坐标系统包括坐标系、投影、原点、单位，两个轴，还有一个起始偏移量。为使不同的地图数据得以发布和集成，在地图预先生成前，应将地图的地理坐标系和投影方式进行统一，转换到地图发布所用的地理坐标系。系统需要支持两种坐标格式的转换，分别是地理坐标系统的经度/纬度（long, lat）和假想的像素空间下的横/纵坐标（X/Y）。

在每个缩放级下，需要实现后从像素空间坐标到地图贴片坐标的转换。假如地图贴片坐标用  $x$  和  $y$  表示，那么需要实现的算法是： $Xtox: function(X, zoom)$  和  $Ytoy: function(Y, zoom)$ 。转换成地图贴片坐标后，整个像素空间可以由带有缩放级别和坐标的地图贴片填满。一般情况下，在最大的缩小级别（zoom 18）下，由一幅地图贴片来表示整个地球，其坐标为  $(zoom, x, y) = (18, 0, 0)$ 。



## 6、电子地图数据库结构

电子地图的数据库系统是电子地图系统重要的组成部分之一。它存储了有关地理信息、空间数据、图层信息等重要的信息。其中包含不同种类的数据库表。

地图数据库的表一般可以分为 5 类：地图构成信息表 MAP，地图图层信息表 LAYER，几何信息表 GEOMETRY，空间参考表 SPATIAL\_REFERENCE\_SYSTEM 和空间索引表 INDEX\_SYSTEM。这些表主要用来存储关于空间坐标系的信息，包括数据库中的坐标系的编号(SRID)、标准的坐标系的名称(AUTH NAME)和编号(AUTH SRID)、详细的文字描述(SRTEXT)。该表通过 SRID 与 MAP 表建立连接关系。MAP 表用来存放地图构成信息，包括图层编号(LID)、图层名称(LNAME、坐标系编号(StuD)、空间数据类型(LTYPE)。该表通过 LID 与 LAYER 表建立连接关系。LAYER 表是一个空间特征表，用来存放图层信息，包括图层编号(LID)、几何特征编号(GID)、各类属性信息(<ATTRIBUTES>)。该表通过 GIS 与 GEOMETRY 表建立连接关系。GEOMETRY 表用来存放空间特征的具体几何信息，包括几何特征编号(GID)、空间范围(XMIN、YMIN、X MAX、YMAX)、空间数据(WKB\_GEOMETRY)。其中空间数据是以 WKB(Well-known Binary)的二进制进制形式存放的。INDEX 表存放索引信息，指明空间特征落在哪一个单元格里，包括几何特征编号(GID)、单元格编号(CELLID)。

### 第三章 移动环境下面向电子地图服务系统模型

移动环境下面向电子地图的服务系统主要由客户端系统和服务器系统两大部分组成。因此开发团队主要分为服务器系统开发与客户端系统开发。论文作者主要负责客户端系统的设计与开发工作。

系统的整体结构如图 3-1 所示：

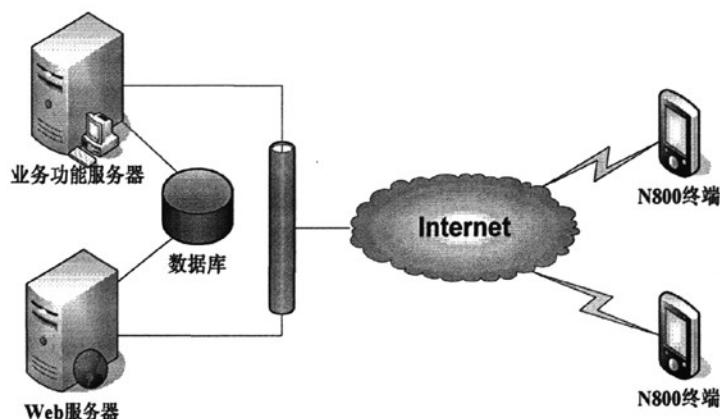


图 3-1 系统整体结构示意图

服务器部分主要由业务功能服务器、Web 服务器和 GIS 数据库组成。业务功能服务器和 Web 服务器部署在同一台物理服务器上，两个服务器使用同一个数据库，通过数据库建立彼此间的联系。

客户端采用诺基亚 N800 手持终端。N800 通过无线局域网（WLAN）与服务器进行交互，完成用户诸如查询、导航定位、踪迹显示等电子地图服务。该电子地图服务在客户端无需使用网络浏览器，直接使用电子地图软件实现。

N800 终端除了直接利用客户端地图软件实现电子地图服务外，还可以通过浏览器查看服务器端搭建的 SNS 网站，网站用户与移动终端用户相对应，每个移动用户都可以登录到自己的 SNS 社区里查看自己的信息，并且可以通过浏览器使用其它电子地图服务。

PC 机也可以通过浏览器浏览 SNS 网站的内容。

#### 3.1 服务器端组成及相关原理

服务器端主要包括 web 服务器和业务功能服务器。Web 服务器上部署一个 SNS 网站，该网站用来管理用户基本信息和用户社会关系信息；业务功能服务器

用来实现系统所涉及的各个业务功能,该服务器完成组件发布、服务组合的功能,为客户端提供各种业务。这些业务包括:

- ✓ POI 业务
- ✓ Travel Plan 业务
- ✓ Travel Blog 业务
- ✓ Tracing 业务

这些业务由服务器端的业务组件 (Service Component) 构成, 包括以下组件:

表 3-1 服务器端的业务组件

组件	功能
POI Manager	接收客户端请求, 根据客户端的查询、添加条件, 生成查询、添加数据库的关键词; 对数据库操作的返回结果进行格式化处理并返回信息给客户端。
POI DB Manager	根据 POI Manager 生成的查询条件对数据库进行添加、查询、更新和删除操作
Travel Plan Manager	接收客户端请求, 根据客户端的查询、添加条件, 生成查询、添加数据库的关键词; 根据一定算法进行 PLAN 拼接; 对数据库操作的返回结果进行格式化处理并返回信息给客户端。
Plan DB Manager	根据 Travel Plan Manager 生成的查询条件对已有的旅游路线信息或旅游景点的 Blog、POI 进行查询; 查询 Plan 所有者和查询者的社会关系; 保存用户的 Plan 信息
Travel Blog Manager	接收客户端请求, 根据客户端的查询、添加条件, 生成查询、添加数据库的关键词; 对数据库操作的返回结果进行格式化处理并返回信息给客户端。
Blog DB Manager	根据 Travel Blog Manager 生成的不同条件对 Blog 进行搜索; 存储用户上传的 Blog, 并生成相应的 Plan 信息
Tracing Manager	接收客户端的 GPS 信息, 调用 DB Manager 进行存储
DB Manager	对 Tracing 的 GPS 信息进行存储

对于上述 4 个主要业务功能, 服务器端通过服务组合流程实现。服务器端实现以下五个服务组合流程: POI、PLAN、BLOG、Tracing、Authentication。这些流程完成服务的组合, 为客户端提供完整的业务。

### 3.2 客户端系统组成与相关原理

客户端系统采用移动网络设备, 硬件部分我们采用诺基亚 N800 移动终端。N800 是诺基亚推出的移动互联网终端 (Internet Tablet)。它采用了基于 Linux 内核 2007 tablet edition 操作系统, 使用 TI OMAP 2420 处理器, 支持 IEEE802.11 网络协议。

该程序在运行于 N800 上的地图软件 Maemo-mapper-1.4.7 的基础上进行开发。Maemo-mapper 是一款专门为 Maemo 平台和 NOKIA N800/810 系列手持终端设计的开源地图软件，它从互联网上的地图仓库（如 OpenStreetMap）下载各种网上地图，实现了电子地图的动态下载与显示功能。因此，从开发角度讲，可以利用 Maemo-mapper 程序原有的地图显示功能，基于该应用程序开发并实现与服务器的通信，并在该地图软件上开发出符合需要的移动电子地图服务。所以，可以把客户端系统分为以下几部分：

- ✓ 通信接口部分
- ✓ 程序各个功能模块
- ✓ 系统整体界面
- ✓ 客户端数据库系统

其中，通信接口为客户端到服务器的接口，用于传送用户请求和查询结果相关的数据；功能模块为不同的电子地图服务，大体分为 4 部分，对应服务器端的各个组合流程组合实现的 4 个主要业务：POI 业务、Travel Plan 业务、Travel Blog 业务、Tracing 业务；系统整体界面为电子地图程序的主界面，该界面的主要部分以 Maemo-mapper 的地图为背景，再辅以必须的功能菜单和功能键，以及地理信息的动态显示功能。

客户端数据库系统是该程序客户端部分重要的组成部分之一。不同于服务器端数据库，由于客户端为资源受限系统，在处理能力和存储能力上和服务器相比都有很大的差距，因此不适宜安装大型数据库系统，并且数据库表单的设计上也与服务器端有很大的不同。数据库表的设计将在客户端系统详细设计中详细介绍。

应用程序采用 C 语言实现，程序的编码及调试过程在 Scratchbox 模拟器上进行。Scratchbox 是一个帮助嵌入式 Linux 应用程序开发的跨平台编译工具，他提供了完整的整合工具链以用来跨平台编译，在 Maemo 开发平台上（Nokia 770, N800, 以及 N810 网络行动装置）使用，这个开发工具支援 ARM 架构以及 x86 架构，在 Maemo 开发平台上（Nokia N770, N800, 以及 N810 互联网终端）使用。

编译及调试通过后通过 Linux 打包程序（dpkg）将源文件打包成可执行文件（.deb），最后将可执行文件安装到 N800 上。

### 3.3 客户端系统开发平台及编程环境

#### 3.3.1 Maemo 开发平台和 Hildon 应用框架

NOKIA N800 移动终端上的应用程序开发需要使用 Maemo 开发工具包，其 UI 框架采用 Hildon Framework 开发完成。

Maemo 是一款为在诺基亚 N770、N800 系列的移动互联网终端上进行应用程序开发的开源的开发平台，其目标是为移动手持终端提供一种应用和创新技术。

该平台为开发商提供了一个强大的基于 Linux 的开发环境和优化的最终用户界面，允许程序开发人员为该类手持设备制作应用软件产品。

Maemo 是构建于桌面开源组件的计算机架构平台该平台基于 GNU/Linux 操作系统和 GNOME 桌面技术[4]。Maemo 向开发伙伴们提供了一个方便实用的开发环境。其全新、优化，且不断演进的 Hildon UI 专为触摸屏手持终端的屏幕大小和典型用途而作了优化。这一开发平台面向那些为移动领域开发应用和新技术的创新开发伙伴和创新型机构。用 maemo 平台能方便地对现有的桌面解决方案进行移动化。Maemo 支持该平台之上的多种业务模型，包括专属应用的分发等。

Maemo 平台向开发伙伴们提供了既强大又方便的开发、构建和测试环境。该平台向你提供了与桌面环境相似的功能。因为 maemo 平台是一个标准的开发环境，移植现有应用非常方便。手持终端上的开发得到了简化，从而增强了开发效率。应用测试和调试也很方便，因为开发环境所运行的软件与目标终端是一样的。基于 Maemo 的终端运行于 ARM 架构之上，而软件开发工具包目前为一些桌面 Linux 发行版而提供的。如 Debian 和 Ubuntu。

运行于桌面机的 Maemo 开发环境被称为 maemo SDK。它只能被安装和运行于 Linux 操作系统。在诸如 Windows 这样的其它操作系统上，可以使用一个 VMWare 镜像以提供 Linux 工作环境。Maemo SDK 在 GNOME/Linux 桌面系统上创建一种沙箱模式的 maemo 开发环境。原则上，这种砂箱环境是建立在被称为 Scratchbox 的工具上。在多数情况下这种环境表现为终端上的操作系统，但增加了一些开发工具。这意味着：开发过程与普通桌面 Linux 和 GNOME/GTK+ 开发十分相似，而其独特的嵌入式开发功能，如交叉编译等，都是由 Scratchbox 透明地处理的。

Maemo 应用程序被建立在 Hildon 框架顶层。简单的应用程序单独连接 Hildon 库、GTK+、Glib 和 LibOSSO 来使用 GUI 元素。更复杂应用程序

依据其需要使用其它服务。例如，某应用程序可能连接 GStreamer 获得多媒体通道，连接 libcurl 获得 HTTP 通道。

Hildon 开发库是 MID 和应用程序产生 GUI 的库的集合。它们基于 GTK+ 库扩展，通过 GTK 窗体部件设计界面。它们是应用程序框架的基本成员，包括 Hildon 窗体部件集合（窗体，菜单，工具条等等），Hildon 桌面，Hildon 文件管理，通用状态条，等等。

Hildon 包含三大库：libhildon、libhildondesktop 和 libhildonfm。

- ✓ Libhildon 包含窗体部件集合的库，用以应用程序的图形界面。窗体部件来自 GTK+ 开发库

Hildon 桌面是手持设备桌面环境。典型 Hildon 桌面包括任务导航器、状态条（例如显示电池或无线信息）和系统菜单，等等。相应的 libhildondesktop 提供必须的项建立自定义桌面对象。在 Moblin 中，主屏幕和 Marquee 开发都使用

libhildondesktop 库。下表为部分 Hildon 桌面组件：

表 4-1 Hildon 桌面组件

名称	描述
HildonDesktopHomeItem	用以 HildonHomeArea 子类的基类
HildonDesktopPanelItem	控制项起源的基类
TaskNavigatorItem	任务导航器起源的基类
StatusBarItem	状态条起源的基类
HildonDesktopPopupWindow	开发桌面弹出窗口的基类
HildonDesktopPopupMenu	开发桌面弹出菜单的基类

- ✓ Hildon 文件管理基于 Gnome VFS，提供统一机制管理存放在本地驱动器和通过 URL 访问的远程容器中的文件。Hildon 文件管理库包括一系列易用、内建对象，像标准 Hildon 文件选择对话框、文件容器窗体部件及用来获取文件信息和属性的对象。

表 4-2 Hildon 文件管理组件

名称	描述
HildonFileSelection	选择文件的容器部件
HildonFileSystemModel	显示文件系统结构的窗体部件
HildonFileChooserDialog	基于 GtkFileChooser 选择文件的窗体部件
HildonFileDetailsDialog	基于 GtkDialog 显示详细文件信息的对话框窗体部件

HildonFilesystemInfo	查询系统相关文件信息的对象
----------------------	---------------

下图说明应用程序、 Hildon 、 GTK+ 、 X lib 和其他 Linux 核心组件在 Moblin 项目中的关系:

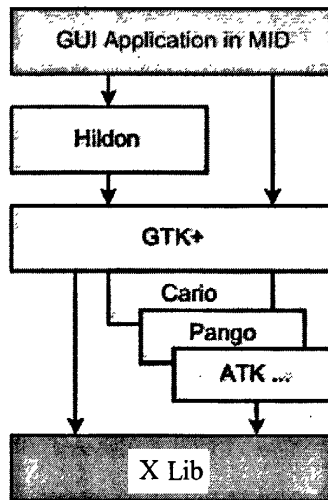


图 4-1 Hildon 程序结构

应用程序可以调用 Hildon API 函数创建特定 Hildon 对象。例如，一个 Hildon 应用程序请求调用“hildon\_window\_new()”函数为自己创建顶层窗口对象。Hildon 应用程序框架处理请求，并调用底层 GTK 函数创建窗口并呈现，等等。有很多其它选择，应用程序可持续调用 GTK 函数处理 UI 对象。例如，Hildon 应用程序可以调用“hildon\_window\_new()”函数创建按钮控件。在 Hildon 应用程序框架上，按钮将通过 Hildon 呈现，但所有底层操作保持无变化的 GTK。

### 3.3.2 GTK+

GTK+是用 C 语言编写的用于开发图形界面程序的函数库。GTK+来源于 GIMP(GNU Image Minipulation Program 即 GNU 图像处理程序)。GTK+在 GDK(GIMP Drawing Kit 即 GIMP 绘图包)基础上创建，对它进行封装。GTK+简单易用，它设计良好，灵活而富有扩展性。它是自由软件，这意味着它不仅开放源代码，而且还可以免费使用。由于它使用 C 语言作为其开发语言，而 C 语言是跨平台的，因此 GTK+几乎可以在任何操作系统上使用。

GTK 实质上是一个面向对象的应用程序接口 (API)。尽管完全用 C 写成的，

但它是基于类和回调函数 (指向函数的指针) 的思想实现的。

还有一个名为 GLib 的第三个组件, 包含一些标准函数的替代函数, 以及一些处理链表等数据结构的函数等。这些替代函数被用来增强 GTK 的可移植性, 因为它们所实现的一些函数在其它 Unix 系统上未实现或不符合标准, 比如 `g_strerror()`。一些是对 libc 的对应函数的增强, 比如 `g_malloc()` 具有增强的调试功能。

在 2.0 版中, GLib 又加入这样一些新内容: 构成 GTK 类层次基础的类型系统 (type system), 在 GTK 中广泛使用的信号系统, 对各种不同平台的线程 API 进行抽象而得的一个线程 API, 以及一个加载模块的工具。



## 第四章 客户端系统需求分析和总体设计

### 4.1 需求分析

面向电子地图的服务除了众所周知的地图显示和位置查询功能外,还针对不同人群提供不同类型的专有服务。本系统提供的面向移动环境的电子地图服务针对外出旅游人群,可以为旅游的人群提供旅游线路规划、旅游博客查看与编写、用户兴趣点查找与管理、用户旅游踪迹跟踪等特色服务。

通过与服务器的交互,移动用户还可以与其他用户分享旅游经历和感想。用户可以在服务器端提供的 SNS 社区里对旅游的景点和路线进行评论,添加自己的兴趣点(POI)到服务器的 GIS 数据库中,实现地理信息数据共享。

#### 4.1.1 需求分析

##### 1 功能需求:

客户端系统主要可以分为以下几大功能:

##### ✓ 端系统地图显示功能、

在客户端屏幕上显示电子地图,可以进行地图的拖拽、显示不同缩放级的地图,并对地图数据进行管理,包括地图图片载入、存储,地图区域调整等。

##### ✓ 用户兴趣点管理功能

电子地图服务中某个位置或地点被称作用户兴趣点(POI)。用户兴趣点服务包括对某个兴趣点进行查询过滤、添加、存储管理等。用户兴趣点(POI)的信息包括:该 POI 的名称、类别、经纬度、描述信息、图片等。

##### ✓ 用户踪迹显示

该业务记录用户旅行轨迹的经纬度信息。客户端通过 GPS 以一定的时间频率记录用户的经纬度信息,并在客户端电子地图上实时显示该踪迹线路,该线路可供用户规划使用。客户端记录的踪迹信息可以批量上传给服务器,上传后的踪迹信息可供其他用户使用。

##### ✓ 移动 BLOG 查看与添加

Blog 业务为用户提供游记记录、分享以及查阅的功能。该业务可通过客户端进行 Blog 创建和编辑、Blog 上传、Blog 搜索下载,也可通过服务器的 SNS 网

站进行 Blog 的创建、编辑和查看。

客户端根据多维查询条件，搜索自己和他人的 Blog 信息进行查阅。

## 2 接口需求：

- ✓ 客户端与服务器交互接口
- ✓ 与本地数据库的接口

## 3 数据需求

客户端需要维护的数据信息，包括 POI 属性信息，用户基本参数，好友列表信息。

### 4.1.2 业务功能描述

#### 4.1.2.1 用户兴趣点 (POI) 业务

##### 1. POI 搜索：

客户端向服务器端请求查询 POI 信息，并向服务器端发送相关查询条件。服务器端根据查询条件搜索相应的 POI 信息，并按照客户端要求的格式将 POI 信息返回。POI 信息包括：名称、类别、经纬度、描述信息、图片。

具体的搜索方式有：

网络搜索，包括以下三类搜索：

按地名搜索

客户端以地名为查询条件，向服务器端请求查询名称中包含某一地名的 POI。

按地名、类别和范围搜索

客户端以地名、类别和范围为查询条件，向服务器端请求查询某一地理位置附近一定范围内的某一类别的 POI。服务器端需要有 GIS 功能模块来将地名转换为该地理位置对应的经纬度信息，并需要将以距离表示的范围转换为一个经纬度范围。

按经纬度、类别和范围搜索

客户端以经纬度、类别和范围为查询条件，向服务器端请求查询某一地理位置附近一定范围内的某一类别的 POI。服务器端需要将以距离表示的范围转换为一个经纬度范围。客户端经纬度的获得方式有两种：点击地图上的某点获得该点经纬度或者有 GPS 模块获得当前经纬度。

本地搜索：

客户端将每次从服务器端获得的 POI 信息存储在本机上，可以在本机上查询这些 POI 信息。

## 2. POI 搜索结果排序：

在以上四种 POI 搜索的基础上，根据好友推荐和公众推荐等 Social Network 属性对搜索结果进行排序，包括以上两类：

### 一度二度好友推荐优先

如果用户在客户端选择“好友推荐优先”的选项，则服务器端将依据该用户一度二度好友的推荐评价对 POI 搜索结果进行排序。

### 公众推荐优先

如果用户在客户端选择“公众推荐优先”的选项，则服务器端将依据所有用户的推荐评价对 POI 搜索结果进行排序。

## 3. POI 添加：

客户端将某个 POI 的名称、经纬度、所属类别、描述信息、图片、行车路线等信息上传到服务器。服务器将该信息存储在一个临时表中，经过一定的认证机制后决定是否将该 POI 信息添加到正式的 POI 库中。

### 4.1.2.2 Travel Plan 业务

Travel Plan 业务为用户提供旅行规划功能。该业务的功能可以分为热点路线推荐、路线搜索、景点搜索、PLAN 制定和重新规划 PLAN。

#### 1. 热点路线推荐

服务器根据客户端所在地区推送该地区目前的热点路线。热点路线的具体信息待定。

#### 2. 路线搜索

客户端以一个或多个景点地名为查询条件，向服务器查询包含这些景点的路线，有以下几种情况：

如果有多条完全匹配景点名的路线，服务器端向客户端返回两条路线查询结果：公众打分最高的路线，和用户好友打分最高的路线。

如果没有完全匹配的路线，则对含有这些景点的路线进行拼接并返回拼接后的路线。

如果没有路线含有这些景点名，则返回这些景点名匹配的 POI。

在返回结果时，服务器将搜索 Plan 所有者和查询者之间的社会关系（一度、二度或陌生人），从而提高信息的可信度。

#### 3. 景点搜索

通过客户端输入某个景点的名称，服务器将返回该点已有的游记，供用户制

定 PLAN 时参考。

#### 4. 制定 PLAN

客户端的用户制定自己的旅游计划。PLAN 的信息包含 PLAN 的名字、用户名、PLAN 包含的景点名、各景点的经纬度信息、各景点的旅行时间。用户制定自己的 PLAN 可以是以服务器查询或推送的路线为基础的,也可以是用户自己新建的。制定好的 PLAN 可以上传给服务器。该 PLAN 可以作为书写 Blog 的框架和基础 (Blog 添加由 Blog 服务完成)。

#### 5. 重新规划 PLAN

当 Tracing 服务所记录的经纬度和时间与原来制定的 PLAN 不相同,客户端将提醒用户是否重新规划制定新的 PLAN。“重新制定 PLAN”的流程与“制定 PLAN”一样。新 PLAN 中应当包含原 PLAN 中已经旅游过的景点。

### 4.1.2.3 Travel Blog 服务

Travel Blog 业务为用户提供游记记录、分享以及查阅的功能。该业务可通过客户端进行 Blog 创建和编辑、Blog 上传、Blog 搜索下载,也可通过 SNS 网站进行 Blog 的创建、编辑和查看。

#### 1. Blog 创建和编辑

客户端以用户制定的 PLAN 为框架,生成 PLAN 中包含的景点的 Blog,由用户进行编辑。客户端 Blog 的信息包括 Blog 主题、所属用户、标签、文字内容、Blog 编辑时间、图片、是否公开等。

#### 2. Blog 上传

客户端将属于某条 PLAN 的所有 Blog 批量上传给服务器。

#### 3. Blog 搜索下载

客户端根据多维查询条件,搜索自己和他人的 Blog 信息进行查阅。

#### 4. SNS 网站 Blog 操作

通过客户端上传的 Blog 信息可在 SNS 网站上进行查看、评论和打分。

通过 SNS 网站可编辑自己已有的 Blog 信息,或创建新的 Blog 信息。

通过 SNS 网站搜索相关的 Blog 信息,包含两种搜索方式:(1)输入关键字对 Blog 全文进行搜索;(2)通过标签对 Blog 信息进行分类查找。

### 4.1.2.4 Tracing 业务

Tracing 业务记录用户旅行轨迹的经纬度信息。该服务的功能有 Tracing 记录、GPS 信息匹配并提示、Tracing 上传。

#### 1. Tracing 记录

客户端 GPS 模块以一定的时间频率记录用户的经纬度信息, 该信息供客户端重新规划 PLAN 使用。

### 2. GPS 信息匹配并提示

客户端 GPS 模块周期将用户当前的经纬度信息与用户制定的 PLAN 进行匹配, 如果匹配不成功, 即提示用户。

### 3. Tracing 上传

客户端记录的 Tracing 信息可以批量上传给服务器端。可以在服务器端显示该用户的 Tracing 轨迹。

## 4.1.3 性能和运行环境需求

由于客户端采用 NOKIA N800 网络终端, 故开发过程须在集成有 Maemo SDK 的 Debian Linux 上进行。因此选用集成 Maemo SDK 的 Ubuntu Linux 6.10。Maemo API 版本为 3x。

软件开发环境:

- ✓ Ubuntu Linux 6.10 (集成 Maemo SDK)
- ✓ Scratchbox 模拟器
- ✓ Xephyr X Server
- ✓ Maemo-mapper-1.4.7 源代码

**Scratchbox 模拟器:** Scratchbox 是交叉编译工具包, 设计用于使嵌入式 Linux 应用程序开发变得更简单。它还提供了一整套用来集成和交叉编译整个 Linux 发行的工具。Scratchbox 最重要的功能可能是它使用一个仿真程序在主机系统上运行目标二进制文件, 即使它们针对不同的架构; 这样就可以无缝地进行交叉开发。

**Xephyr X Server:** 它可以当作一个 X client 连接到实际系统中, 这对于运行程序是非常必要的。

## 4.2 总体设计

客户端系统基于 maemo-mapper1.4.7 开源地图软件进行电子地图服务开发。Maemo Mapper 是一款专门为 Maemo 平台和诺基亚 N770/N800/N810 定制设计的电子地图软件, 它从从互联网上的地图仓库 (如 OpenStreetMap) 中下载各种地图。

论文作者利用 **Maemo-mapper** 程序原有的地图显示功能, 在该应用程序的基础上开发并实现与服务器的通信, 并在该地图软件上开发出符合需要的移动电子地图服务。

通过对该软件的改写, 使其完成需求分析中的各项功能, 并通过无线局域网 (WLAN) 与服务器进行数据交互。

通信接口为客户端到服务器的接口, 用于传送用户请求和查询结果相关的数据。

#### 4.2.1 端系统整体结构

由图 4-2 所示, 客户端系统可分为: 用户界面、POI Guide 模块、Travel Plan 模块、Travel Blog 模块、Tracing 模块、本地数据库系统、接口模块。

其中, 4 个功能模块为不同的电子地图服务对应服务器端的各个组合流程组合实现的 4 个主要业务: POI 业务、Travel Plan 业务、Travel Blog 业务、Tracing 业务; 系统整体界面为电子地图程序的主界面, 该界面的主要部分以 **Maemo-mapper** 的地图为背景, 再辅以必须的功能菜单和功能键, 以及地理信息的动态显示功能。

用户界面采用 **Hildon** 框架, 4 个功能模块对应服务器端提供的相应的服务, 这些功能通过 **gtk+** 事件循环机制通过信号调用启动, 客户端系统与服务器的通信接口采用 **SOAP** 协议实现。由于客户端开发采用 C 语言编写, 客户端部分的数据接口也采用 C 语言实现, 故接口模块在实际开发中使用 **gSOAP** 与服务器进行数据交互, 从服务器端传送过来的数据保存到本地数据库中。客户端本地数据库采用 **sqlite3** 数据库系统。

下图为客户端系统模块示意图:

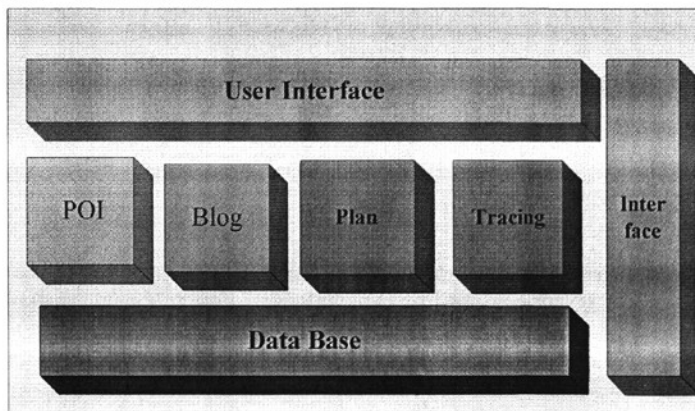


图 4-2 客户端系统模块

## 4.2.2 客户端系统 UI 设计

电子地图的界面通常由图幅窗口、索引图窗口、图幅名称列表框、热点名称列表框、地图名称条、系统工具条、伴随窗口、信息窗口、其他信息输入或输出窗口组成。这些界面组成要素有些是永久性的，有些是临时性的，也有些是用户通过交互操作自主选择。

由于客户端系统基于 `maemo-mapper1.4.7` 开源地图软件进行开发，因此以该开源软件的地图界面为基础，增加必要的功能菜单与工具条。由于客户端显示区域有限，所以，界面的设计力图做到界面简洁易于操作。具体 UI 结构如下图所示：

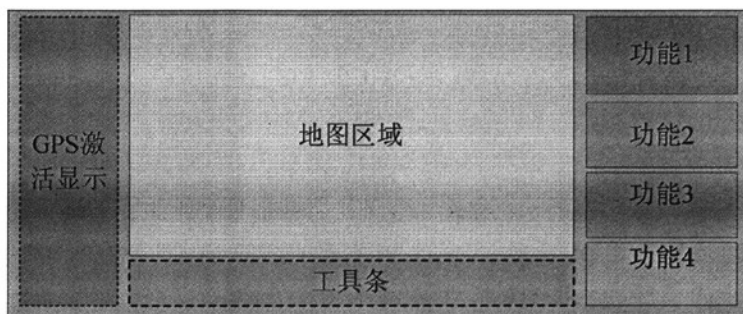


图 4-3 客户端界面布局

客户端主界面在系统中负责显示友好的操作界面，通过主界面上的按钮或菜单项启动程序的各个功能部分，引导用户操作。

主界面包括：菜单项、功能启动按钮、工具条、以地图作为程序的背景。

程序界面布局为：（以 N800 屏幕为基准）

中心：地图显示区，主要显示地图以及查询出来的 POI 和旅游路线；

下方：快捷工具栏，可快速启动比较常用的功能，例如：Add Blog、地图分辨率调整、POI 查询等快捷功能按钮。

右侧：程序主要功能菜单按钮，主要有 POI Guide、Travel Plan、Travel Blog、Tracing 4 个功能。

左上方：地图相关功能调整，例如：地图图片下载、地图分辨率调整等。

左侧：GPS 激活显示区，当连接到 GPS 设备时，该部分区域出现，记录 GPS 设备连接状态和连接时间。通常情况下该部分区域为隐藏状态。

### 4.2.3 界面功能设计

#### 1、功能启动按钮。

**Travel Plan 功能启动：**用户通过点击主界面上“Travel Plan”按钮，即进入 Travel Plan 功能复选界面，此时出现“Recommend Plan”、“Plan Search”、“My Plan”、“Searched Plan” 选择菜单，用户通过点击相应的按钮进入相应的功能界面。

**Travel Blog 功能启动：**用户通过点击主界面上“Travel Blog”按钮，即进入 Travel Blog 功能复选界面，此时出现“Add Blog”、“View Blog”、“View Setting”、“Search Blog” 选择菜单，用户通过点击相应的按钮进入相应的功能界面。

**POI Guide 功能启动：**用户通过点击主界面上“POI Guide”按钮，即进入 Travel POI 功能复选界面，此时出现“Search Location”、“Search POI”、“Clear POI” 选择菜单，用户通过点击相应的按钮进入相应的功能界面。

**Tracing 功能启动：**用户通过点击主界面上“Tracing”按钮，即进入 Tracing 功能界面。

#### 2、工具条。

**工具条呈现及隐藏：**程序开始运行时出现工具条，经过特定时间段后工具条自动隐藏。用户在地图界面范围内双击鼠标可再次将工具条调出。当用户将鼠标移动到工具条的按钮上面时显示按钮名称提示。

**Login 功能：**用户通过点击工具条上“Login”按钮，即进入 Login 功能界面。

**Network Search 功能：**用户通过点击工具条上“Network Search”按钮，即进入 Network Search 功能界面。

**Zoom In 功能：**用户通过点击工具条上“Zoom In”按钮，即可实现 Zoom In 功能，该功能对地图进行放大，最大为 level 0。

**Zoom Out 功能：**用户通过点击工具条上“Zoom Out”按钮，即可实现 Zoom Out 功能，该功能对地图进行缩小，最小为 level 14

**Go to GPS Location 功能：**用户通过点击工具条上“Go to GPS Location”按钮，地图界面则跳转到 GPS 设定的地点。

**Logout 功能：**用户通过点击工具条上“Logout”按钮，即进入 Logout 功能界面。

#### 3、地图背景。



地图显示功能：将 Travel Plan 表示的路线、POI 信息及其地理位置等在客户端地图上显示。

#### 4.2.4 外部接口设计

客户端系统与服务器的通信接口采用 SOAP 协议实现。

SOAP (Simple Object Access Protocol) 协议是在分散或分布式的环境中交换信息的简单的协议，是一个基于 XML 的协议，它包括四个部分：SOAP 封装 (envelop)，封装定义了一个描述消息中的内容是什么，是谁发送的，谁应当接受并处理它以及如何处理它们的框架；SOAP 编码规则 (encoding rules)，用于表示应用程序需要使用的数据类型的实例；SOAP RPC 表示 (RPC representation)，表示远程过程调用和应答的协定；SOAP 绑定 (binding)，使用底层协议交换信息。

由于客户端代码采用 C 语言编写，所以在接口实现中采用了 gSOAP 作为接口的实现。

gSOAP 编译工具提供了一个 SOAP/XML 关于 C/C++ 语言的实现，gSOAP 利用编译器技术提供了一组透明化的 SOAP API，并将与开发无关的 SOAP 实现细节相关的内容对用户隐藏起来。gSOAP 的编译器能够自动的将用户定义的本地化的 C 或 C++ 数据类型转变为符合 XML 语法的数据结构。gSOAP 包含一个 WSDL 生成器，用它来为 web 服务生成 web 服务的解释。gSOAP 的解释器及导入器可以使用户不需要分析 web 服务的细节就可以实现一个客户端或服务端程序。

接口文件由 soapClient.c、soapC.c、soapStub.h、stdsoap.c、soapH.h 这几个文件组成。

soapH.h 头文件由 WSDL 解析器生成，各个功能模块与服务器之间的通信由 soapClient.c 文件提供的接口完成。下图为定义的各个模块的接口函数

##### 1. POI 查询接口：

```
SOAP_FMAC5 int SOAP_FMAC6 soap_call_ns1__QueryPOI(struct soap *soap,
const char *soap_endpoint,
const char *soap_action,
char *_gname,
char *_category,
char *_description,
char *_user, char *_pwd,
int _user_USCOREid,
char *_source,
double _longitude,
double _latitude, double _altitude,
double _scale,
int _poinum,
struct ns1__QueryPOIResponse *_param_1)
```

图 4-4 POI 查询接口

接口中包含以下参数:

**soap\_endpoint:** soap 端点中包含了服务器的地址, 以及在服务器上部署的对应的服务流程。

**soap\_action:** 接口文件定义的 soap 端点的行为。

**\_gname, \_category:** 表示用户欲查询的 POI 的名称和分类, 分类由用户指定, 名称为可选项。

**\_description:** 服务器返回的 POI 的描述信息

**\_user, \_pwd:** 用户登录的用户名和密码, 用来向服务器进行服务认证。

**\_user\_USCOREid:** 该 id 为用户在服务器中注册的唯一 id, 用来标识用户身份。

**\_source:** 如果 POI 中包含多媒体数据, 则通过此参数将多媒体数据用 base64 编码后传送到客户端, 如果 POI 中只包含文本数据, 则此项为 NULL。

**\_longitude, \_latitude, \_altitude:** 返回的 POI 的经纬度和高度, 利用此经纬度将 POI 的点标示到电子地图对应的位置。

**\_scale:** 表示查询范围。

**\_poinum:** 与服务器进行对应的 POI 的编号, 只在交互时使用。

## 2. POI 添加接口

```
SOAP_FMAC5 int SOAP_FMAC6 soap_call_ns1__AddPOI(struct soap *soap,
const char *soap_endpoint,
const char *soap_action,
char *category,
char *gname,
int user_USCOREid,
char *description,
char *address,
char *instruction,
char *file_USCOREname,
char *file_USCOREcontent,
int mark,
char *comment,
double longitude,
double latitude,
double altitude,
struct ns1__AddPOIResponse *param_2)
```

图 4-5 POI 添加接口

## 3. BLOG 添加接口

```
SOAP_FMAC5 int SOAP_FMAC6 soap_call_ns1__QueryBLOG(struct soap *soap,
const char *soap_endpoint,
const char *soap_action,
int memberID,
int friendID,
char *planname,
char *sitename,
char *subject,
char *date,
char *tag,
struct ns1__QueryBLOGResponse *param_5)
```

图 4-6 BLOG 添加接口

#### 4. 认证接口

```
SOAP_FMAC5 int SOAP_FMAC6 soap_call_ns1__Authenticate(struct soap *soap,
const char *soap_endpoint,
const char *soap_action,
char *nickname,
char *pwd,
struct ns1__AuthenticateResponse *param_10)
```

图 4-7 认证接口

认证接口中的 `_nickname` 和 `_pwd` 为用户登录用户名和密码。

#### 4.2.5 功能模块划分

根据项目的功能需求，客户端系统主要划分为 4 个功能模块：

- ✓ POI 模块
- ✓ Travel Plan 模块
- ✓ Travel Blog 模块
- ✓ Tracing 模块

这 4 个模块分别对应服务器端提供的 4 个业务。

POI 模块中，用户通过客户端向服务器端请求查询 POI 信息，并向服务器端发送相关查询条件。服务器端根据查询条件搜索相应的 POI 信息，并按照客户端要求的格式将 POI 信息返回。POI 信息包括：名称、类别、经纬度、描述信息、图片。

Travel Plan 模块实现为用户提供旅行规划。该模块的功能可以分为热点路线推荐、路线搜索、景点搜索、PLAN 制定和重新规划 PLAN。客户端以一个或多个景点地名为查询条件，向服务器查询包含这些景点的路线。

**Travel Blog** 模块实现为用户提供游记记录、分享以及查阅的功能。该模块可通过客户端进行 Blog 创建和编辑、Blog 上传、Blog 搜索下载等功能。

**Tracing** 模块负责记录用户旅行轨迹的经纬度信息。客户端通过 GPS 以一定的时间频率记录用户的经纬度信息，并在客户端电子地图上实时显示该踪迹线路，该线路可供用户规划使用。客户端记录的踪迹信息可以批量上传给服务器，上传后的踪迹信息可供其他用户使用。

在各个功能模块的具体开发中，论文作者主要负责 POI 模块的设计开发与功能实现。

## 第五章 客户端系统 POI 模块详细设计与实现

POI 服务作为电子地图位置服务的一种，该模块是电子地图客户端系统的重要组成部分之一，是移动环境电子地图服务中所不可缺少的功能。因此，该模块的设计与实现在客户端系统中显得尤为重要。

### 5.1 POI 模块详细设计目标和思想

在传统电子地图 POI 查询的基础上增强了移动性、交互性与离线操作特性，除了传统的 POI 查询之外，增加了本地 POI 显示，上传 POI 到服务器，查看好友对某 POI 的评论等等。

本地 POI 显示可将用户已查询的 POI 保存到客户端系统数据库中，在网络不可用时也可在地图上显示。

在客户端显示的 POI 除了具有名称、分类、经纬度、描述等基本信息外，还加入了好友对该 POI 的评价，包括打分、支持度等等，这样更体现了真实性和用户之间的互动性。

系统中的 POI 添加功能允许用户添加自己的 POI 数据到网络服务器中，经过服务器端的认证流程后，可以永久保存到服务器 GIS 数据库中，使得每一个用户都可以完善网络数据库，为其他用户提供便利。这也是 web2.0 思想在电子地图领域的应用。

### 5.2 POI 模块详细设计

考虑到 POI 模块的功能与客户端系统的整体结构，将 POI 模块分为以下几个部分：C\_MainFrame, C\_Travel\_POI, POI MessageSent, POI DataBase。其中，C\_Travel\_POI 中包括两部分：MessageSent、POI\_Process。

C\_MainFrame 用来接收用户传来的消息，并将消息传送给 C\_Travel\_POI；C\_Travel\_POI 根据 C\_MainFrame 发送来的消息处理 POI 数据，数据预处理工作由 POI\_Process 完成，需要服务器处理的数据由 MessageSent 将消息发送到服务器，服务器将处理结果再通过 MessageSent 返回到客户端，经过 POI\_Process 将结果发送到数据库或者回送给 C\_MainFrame，以实现 POI 数据的管理。

POI DataBase 存放 POI 相关的数据，这些数据可能由 C\_Travel\_POI 直接处理得到，也可能通过接口从服务器端获得。

下图为 POI 模块示意图：

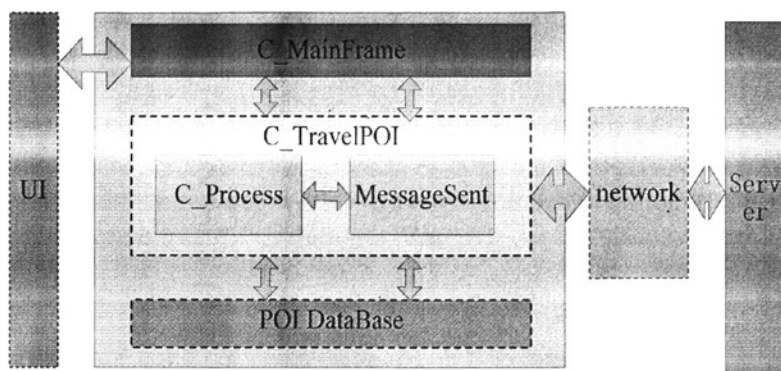


图 5-1 POI 模块示意图

如图，UI 为用户实体，它向 C\_MainFrame 发送消息，通过发送不同的消息类别，启动不同的服务。在 POI 模块中，用户可以发送添加、查询、上传、删除等消息来完成相应的功能，主要实现的功能有：POI 查询（网络/本地）、POI 添加/上传、POI 信息删除、POI 评论信息上传等。

## 5.2.1 POI 查询功能实现

### 5.2.1.1 POI 查询功能界面实现

1. 功能：用户通过 POI 查询功能搜索到想要的 POI 地理位置和相关信息。查询到的 POI 在客户端地图上显示。该功能又分为网络查询、本地查询和点地图查询 3 种。

#### 2. 输入参数：

POI 查询功能界面主要输入参数分为两类，一类是用户输入的参数，另一类是系统获取到的默认参数。其中，用户参数包括：Category（POI 分类），Position（用户查询位置），scale（POI 查询范围），numbers of POIs（POI 显示个数）；系统默认参数包括用户所在当前位置的经纬度、用户 ID 以及历史记录等。这些默认参数不在功能界面中显示，在用户提交具体查询时自动添加。

#### 3. 功能界面

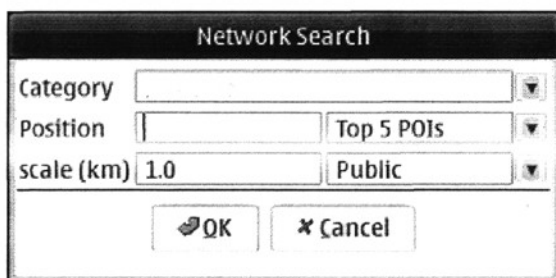


图 5-2 网络查询功能界面

如图 5-2 所示为网络查询功能界面, 查询界面框架由 `gtk_dialog` 构成, 窗体部分采用 3X3 的 `GTK_TABLE` 实现, `GTK_TABLE` 中的每个元素分别代表查询分类、位置、范围、POI 显示个数。

#### 4. 具体说明

- ✓ 当用户点击 `Network Search` 服务时会弹出该界面。
- ✓ 两个输入框分别用来输入用户想要查询的 POI 的分类, 例如: 商店, 加油站... ; 和欲查询的 POI 所在地名称, 例如: 颐和园, 前门... ; `scale` 选项输入所要查询的范围 (以 km 为单位)。
- ✓ 输入后点击“OK”, 程序通过 WLAN 自动连接服务器进行查询; 点击“Cancel”取消查询操作。

#### 5.2.1.2 POI 查询功能消息交互

POI 查询功能分为网络可用与网络不可用两种情况。

1. 当网络可用时, 客户端通过 WLAN 向服务器发送查询请求, 并由服务器返回相应的查询结果到客户端。其 MSC 图如下:

## MSC POI\_search\_network

通过网络查询POI

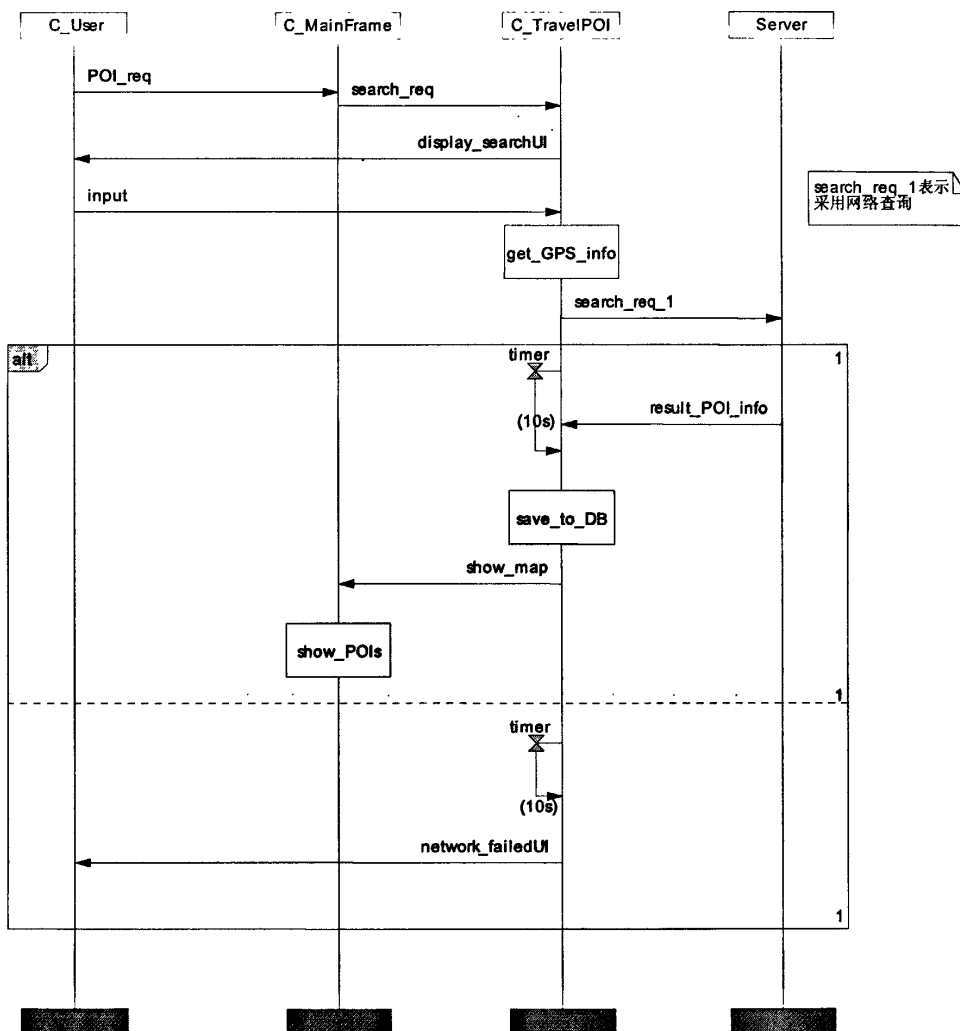


图 5-3 POI 网络查询消息交互

该 MSC 图包括 4 个功能实体: C\_User, C\_MainFrame, C\_TravelPOI 与 Server。其中,

- ✓ C\_User --- 使用客户端的用户
- ✓ C\_MainFrame --- 程序主框架
- ✓ C\_TravelPOI --- POI 处理进程
- ✓ Server --- 后台服务器

用户向 C\_MainFrame 发送 POI\_req 消息提出查询请求, C\_MainFrame 接收到用户请求并将该请求发送到 C\_TravelPOI 处理进程, C\_TravelPOI 发送消息告知用户接受请求并等待用户查询消息 input, 该消息包含查询条件相关信息, 包



括 POI 分类 (category)、POI 所在大致地点 (site) 等, C\_TravelPOI 接收到该信息后通过 WLAN 将其发送给 server, 同时启动超时定时器。

在定时器超时前, 若 C\_TravelPOI 收到 server 发送来的 result\_POI\_info 消息, 则表示此次查询成功, C\_TravelPOI 将查询结果保存到客户端数据库中, 并向 C\_MainFrame 发送目标 POI 显示消息: show\_map; C\_MainFrame 将查询到的 POI 显示到客户端电子地图上。

如果定时器超时, C\_TravelPOI 仍然没有收到 result\_POI\_info 消息, 那么客户端主动断开连接, 并发送网络查询失败消息—network\_failed。

- 2. 当网络不可用时, 客户端通过查找本地的 POI 数据库来获取 POI 信息。其 MSC 图如下:

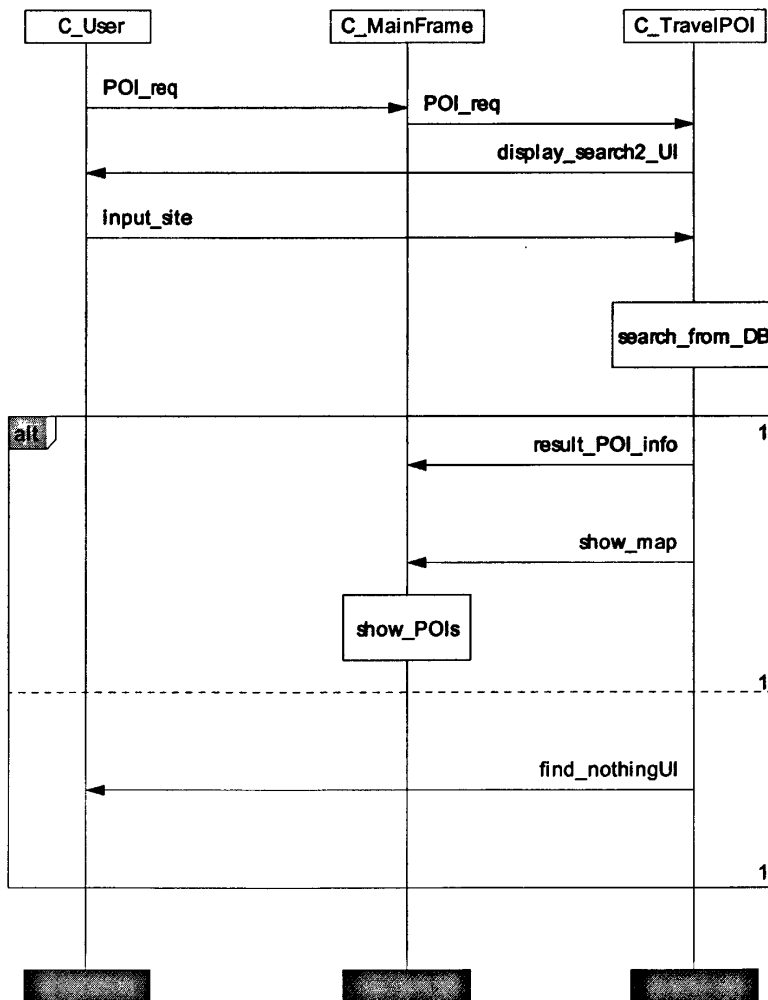


图 5-4 POI 本地查询消息交互

该 MSC 图包括 3 个功能实体：C\_User，C\_MainFrame，C\_TravelPOI。

用户通过点击主程序框架上的菜单项激活 POI Guide 服务，同时启动 POI 查询功能，这时，C\_MainFrame 发送 POI\_req 消息给 C\_TravelPOI 处理进程，C\_TravelPOI 接收到消息后通知用户发送查询信息，这些查询信息包含在 input\_site 消息中。当用户提供了该消息后，C\_TravelPOI 在本地 POI 数据库中搜索，然后将结果数据发送给 C\_MainFrame，最后将 POI 显示在电子地图上。

如果 C\_TravelPOI 在本地数据库中未能搜索到数据，则向主程序发送 find\_nothing 消息，表示搜索不到结果。

### 5.2.1.3 POI 查询状态转移图

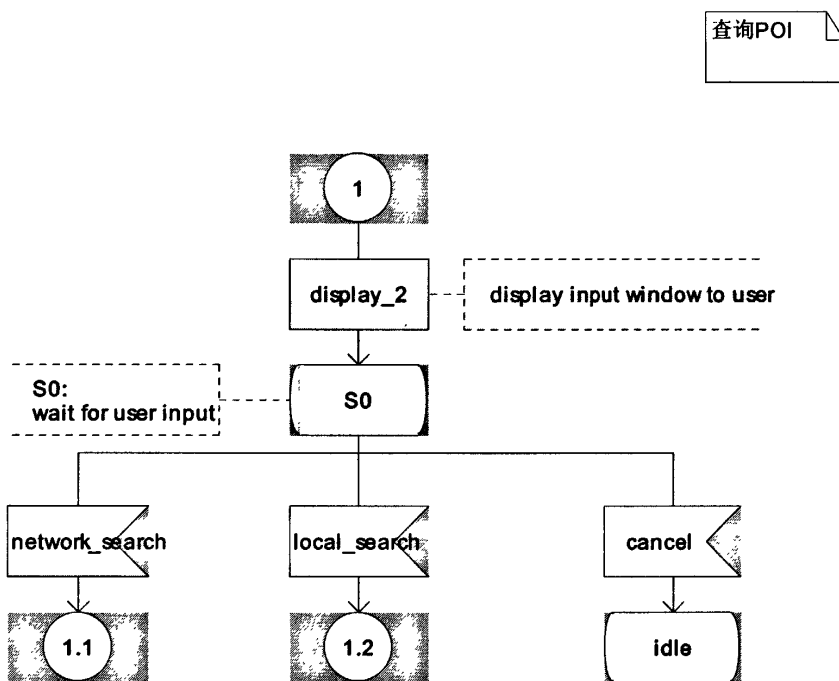


图 5-5 POI 查询状态迁移

- ✓ S0：等待用户输入，接收 3 个输入信号：network\_search, local\_search, cancel;
- ✓ POI 查询功能分为两种情况：当用户所在地网络畅通时，通过 WLAN 连接服务器进行网络查询；当用户所在地网络出现故障或不通时采用本地查询。

通过 WLAN 查询：

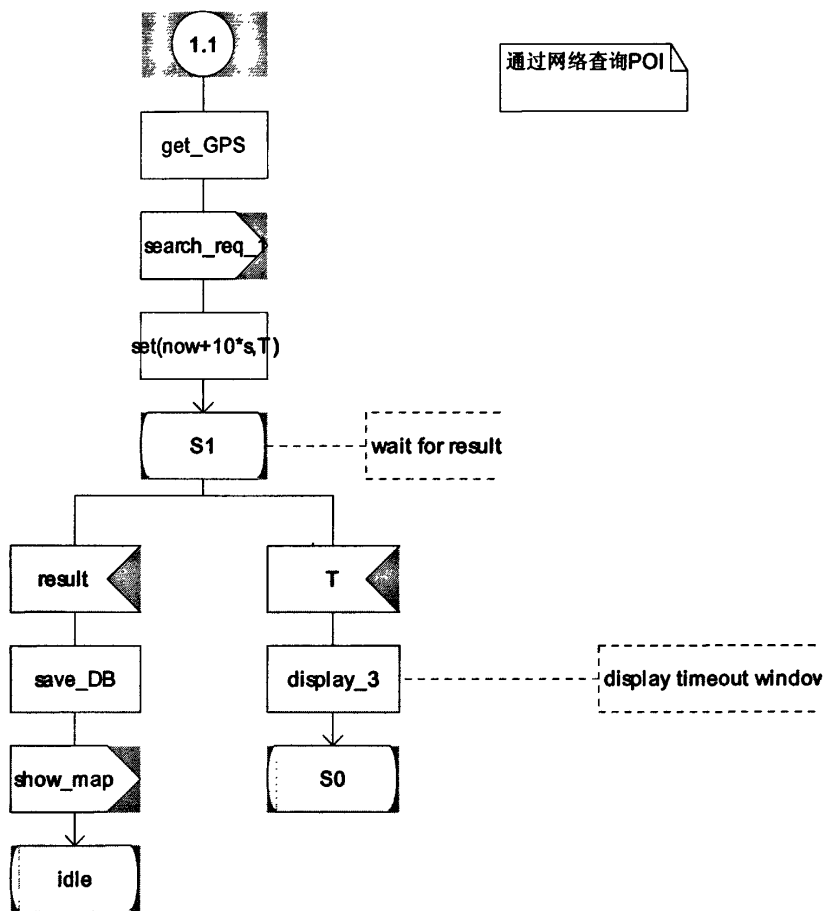


图 5-6 网络查询状态迁移

用户在 S0 状态下接收到 `network_search` 信号后，从 GPS 中获取当前地点的经纬度信息，并将该信息与用户先前的输入一起发送给服务器，同时启动定时器。

若定时器超时之前收到从服务器返回来的查询结果，把该数据存放到客户端数据库，并在地图上标示该点位置。

若定时器超时，显示网络不可用提示窗口，回到 S0 状态。

通过本地查询：

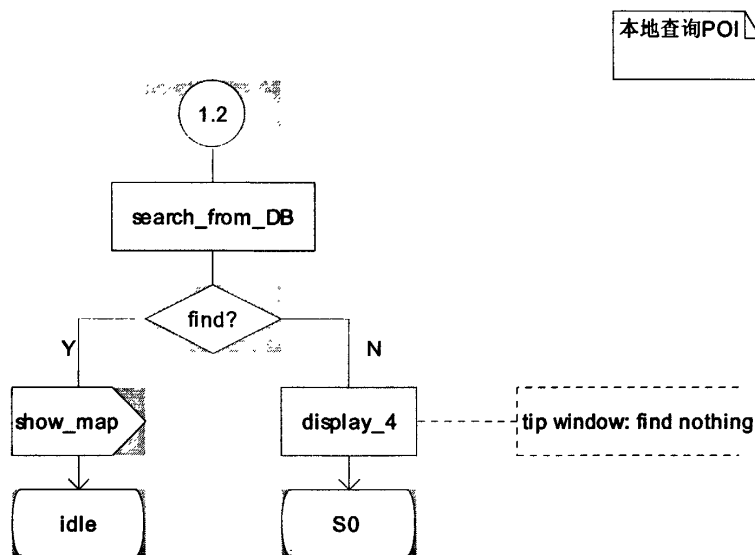


图 5-7 本地查询状态迁移

用户在 S0 状态下接收到 local\_search 信号后,从本地数据库中查询符合条件的 POI,若存在,将这些 POI 显示到地图上;若不存在,显示“找不到”提示窗口,然后回到 S0 状态。

## 5.2.2 POI 添加功能实现

### 5.2.2.1 POI 添加功能界面实现

#### 1. 功能:

用户在某一特定地点记录该地点的 POI 信息,并将该 POI 上传到服务器给他人分享。

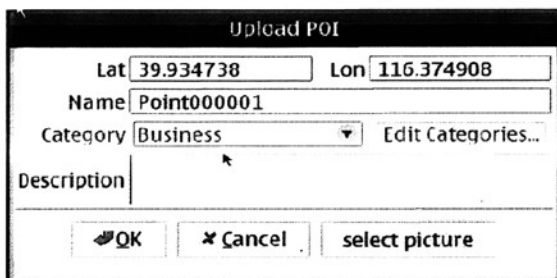
#### 2. 输入参数:

用户输入的参数: Name (POI 名称), Category (POI 分类), Description (POI 相关描述)

系统参数: Lat (纬度), Lon (经度)。

#### 3. 功能界面:

POI 添加功能界面与查询界面类似,其中, Lat 与 Lon 所显示内容由系统自动获取并显示。



The screenshot shows a dialog box titled "Upload POI". It has several input fields: "Lat" with the value "39.934738", "Lon" with "116.374908", "Name" with "Point000001", "Category" with a dropdown menu showing "Business", and a "Description" text area. There are three buttons at the bottom: "OK", "Cancel", and "select picture".

图 5-8 POI 添加界面

#### 4. 具体说明

- ✓ 当用户点击 POI Add 服务时会弹出该界面
- ✓ 用户在 Category 中输入 POI 分类, 在 Description 中输入对该 POI 的描述, 该点的经纬度自动获得并自动写入 Lat 和 Lon 文本框中。
- ✓ 用户点击“add”, 将该 POI 信息写入客户端的数据库并上传到服务器。
- ✓ 用户点击“Cancel”则取消添加操作。

#### 5.2.2.2 POI 添加功能消息交互

用户通过点击主程序框架上的菜单项激活 POI Guide 服务, 同时启动 POI add 子功能, 此时 POI 处理进程向用户提供 POI 信息添加界面, 用户输入该点 POI 信息后点击发送, 将该 POI 信息上传到服务器, 同时该数据也会保存在本地数据库中, 并在地图上显示。

下图为 POI 添加功能的 MSC 图:

## MSC POI\_add

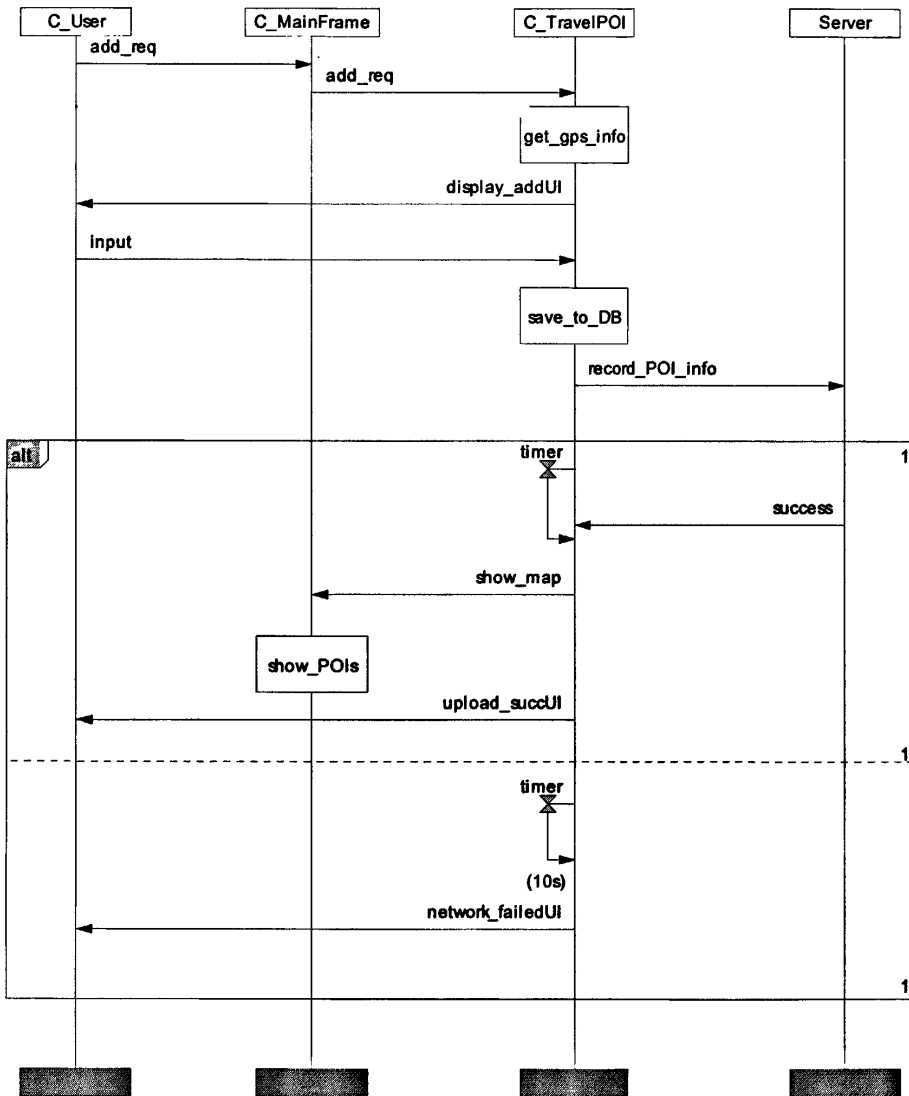


图 5-9 POI 添加消息交互

该 MSC 图包括 4 个功能实体: C\_User, C\_MainFrame, C\_TravelPOI 与 Server。

C\_MainFrame 发送 add\_req 消息到 C\_TravelPOI 启动 POI 添加功能, C\_TravelPOI 首先查询 GPS 设备获取到当前用户的位置信息, 然后通知用户并等待用户添加消息列表数据。

用户将该 POI 信息发送到 C\_TravelPOI, C\_TravelPOI 首先将该信息储存到本地数据库, 然后再将数据通过 WLAN 传送到 server, 同时启动超时定时器

Timer。

当在定时器超时时间内从 Server 端返回添加成功消息 success, C\_TravelPOI 撤销定时器, 并向用户返回上传成功消息 upload\_succUI, 然后将 POI 显示在地图上; 若定时器超时向用户返回上传失败 network\_failed。

### 5.2.2.3 POI 添加状态转移图

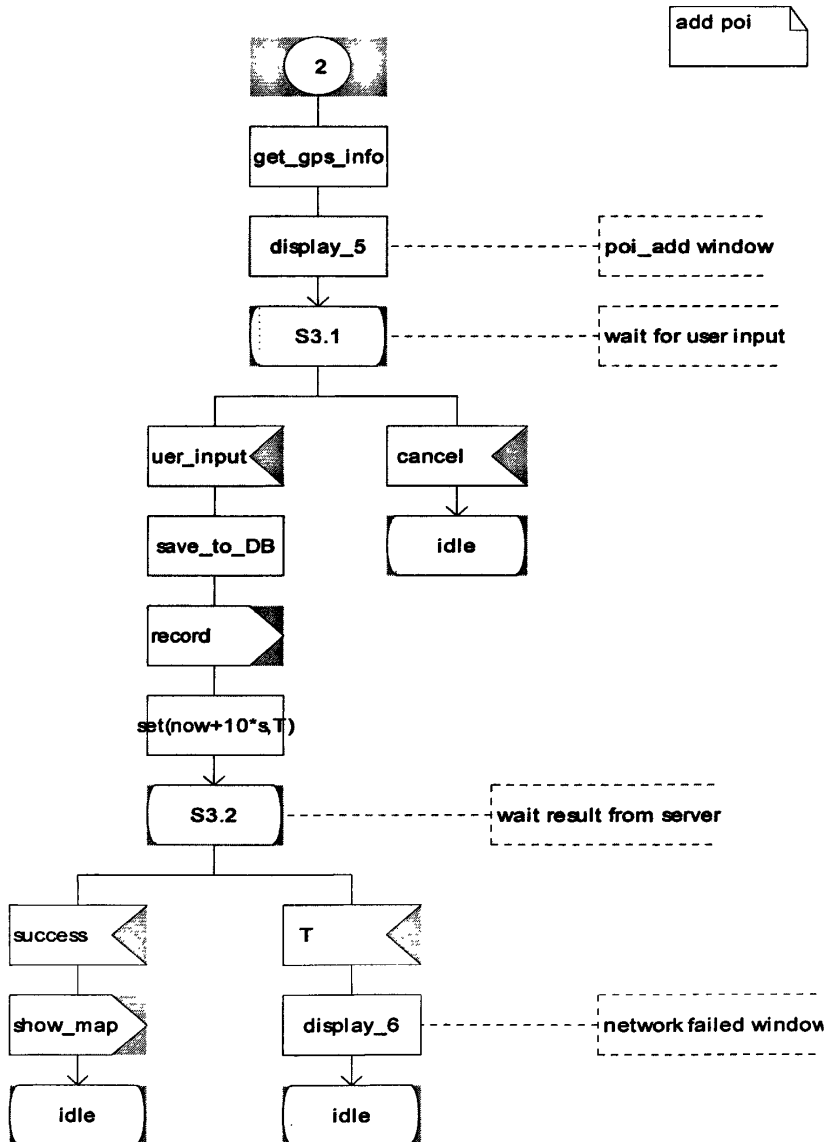


图 5-10 POI 添加状态迁移

用户在 S0 状态下接收到 add 信号后, 从 GPS 中获取当前地点的经纬度信

息，同时显示用户添加 POI 窗口，等待用户输入。

(1) 用户输入后点击“确定”，将用户输入的 POI 信息及从 GPS 中获取的经纬度信息存入客户端数据库，并将该数据发送给服务器，同时启动定时器，进程进入等待结果返回状态 (S3.2)。

若服务器返回成功信号，将该 POI 显示到地图；

若超时，显示网络不可用窗口，进程回到 idle 状态。

(2) 用户点击“CANCEL”，撤消输入框，进程进入 idle 状态。

### 5.2.3 POI 删除功能实现

#### 5.2.3.1 POI 删除功能界面实现

1. 功能：用户将不需要的 POI 信息从客户端数据库删除，同时从地图上删除该 POI 记录。

2. 输入参数：POI 分类。

3. 功能界面

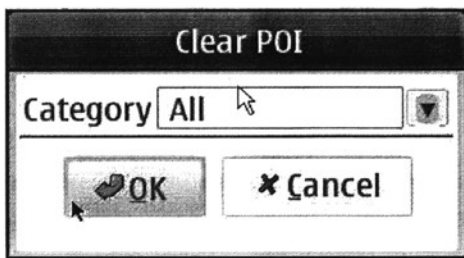


图 5-11 POI 删除界面

4. 具体说明

- ✓ 当用户点击 Delete POI 服务时会弹出该界面。
- ✓ 用户从 Category 中选择要删除的 POI 分类名称，该点其余的 POI 信息（经纬度、描述）会自动显示在文本框中。
- ✓ 用户点击“Delete”将该 POI 从客户端数据库中删除，同时删除地图上对应的 POI。

#### 5.2.3.2 POI 删除功能消息交互

由于服务器端数据库为公用数据库，所以不可能通过客户端删除服务器上的 POI 数据。因 POI 删除功能仅限于删除本地数据库中的 POI 数据。

POI 删除也可以分为临时删除电子地图上显示的数据和永久删除客户端数据库中的数据。



## MSC POI\_delete

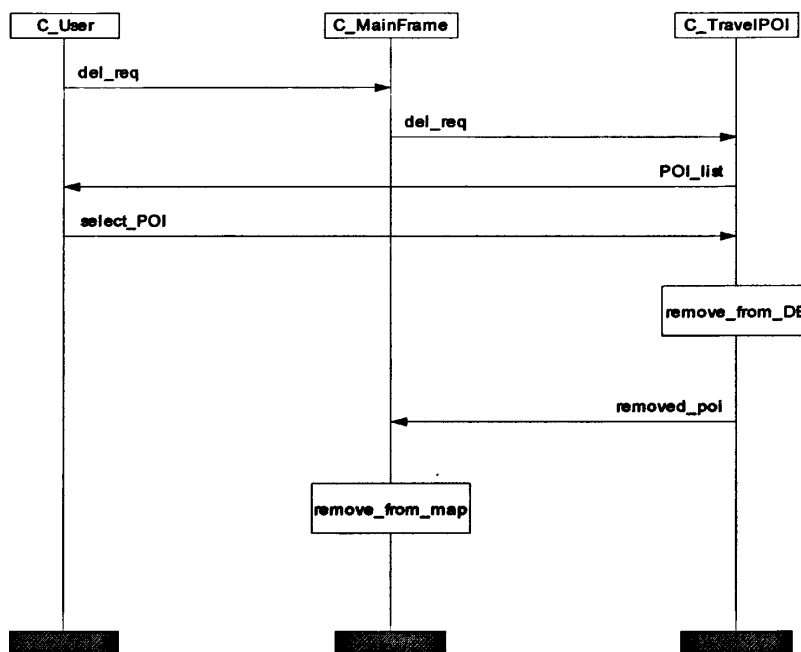


图 5-12 POI 删除消息交互

用户通过点击主程序框架上的菜单项激活 POI Guide 服务，此时，C\_MainFrame 发送 del\_req 消息到 C\_TravelPOI，启动 POI delete 子功能，POI 处理进程向用户提供 POI 信息列表，用户从中选择欲删除的 POI 数据，即向 C\_TravelPOI 发送 select\_POI 消息，C\_TravelPOI 将 select\_POI 消息中对应的 POI 数据从数据库中删除，操作成功后，C\_TravelPOI 回送 removed\_POI 消息给 C\_MainFrame，C\_MainFrame 接收到该消息后清除地图上已经被删除的 POI 点，完成 POI 删除工作。

## 5.2.3.3 POI 删除状态转移图

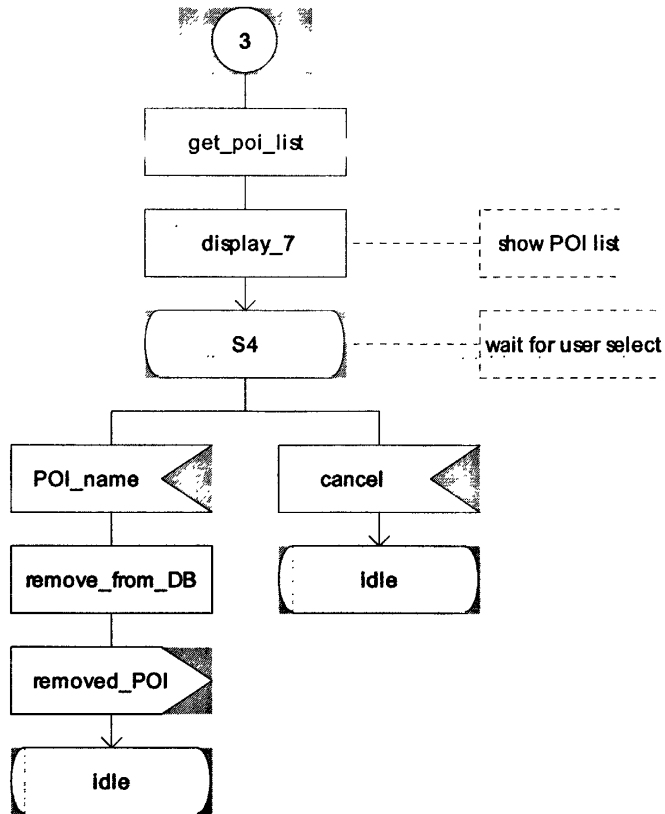


图 5-13 POI 删除状态迁移

在等待删除状态下 (S4)，

(1) 当用户输入欲删除的 POI 名称 POI\_name，从客户端 POI 数据库中将符合删除条件的 POI 删除，删除后回到 idle 状态；

(2) 用户取消该操作，程序不作操作，直接回到 idle 状态。

### 5.3 POI 模块主要数据结构设计

POI 基本信息数据结构：

通常一个 POI 包含有以下基本信息：POI 名称，POI 分类，经度，纬度，POI 描述信息。

```

poi_info
{
    gchar* category;          /* POI 分类 */
    gchar* name;             /* POI 名称 */
    double lat;              /* 纬度信息 */
    double lon;              /* 经度信息 */
}
  
```

```

    gchar* desc;           /* POI 描述信息 */
    gchar* pic_name;      /* 显示图片名称 */
    gchar* pic_content;   /* 图片内容 */
    int mark;             /* POI 评分 */
};

```

描述地图上一个点的信息有横/纵坐标值、时间和海拔：

```

struct _Point {
    guint unitx;
    guint unity;
    time_t time;
    gfloat altitude;
};

```

路径采用链表方式实现，每一个结点为地图上的一个点：

```

struct _Path {
    Point *head;          /* points to first element in array; NULL if empty.
*/
    Point *tail;         /* points to last element in array. */
    Point *cap;          /* points after last slot in array. */
    WayPoint *whead;     /* points to first element in array; NULL if empty. */
    WayPoint *wtail;     /* points to last element in array. */
    WayPoint *wcap;      /* points after last slot in array. */
};

```

DB Manager 回传给 POI Manager 的 POI 信息是一个类数组 POIinfo[]，其结构如下所示：

```

public class POIinfo implements Serializable{
    private String gname;
    private String category;
    private double longitude;
    private double latitude;
    private String description;
};

```

```
private String address;
private String instruction;
private String file_path;
private POIcomment[] comment; //记录一个 POI 对应的评论
}
```

```
public class POIcomment implements Serializable{
private String nickname; //做出评价的用户昵称
private int mark; //评分
private String comment;
}
```

接口

POI Manager 回传给客户端的 POI 信息是一个类数组 POIres[], 其结构如下所示:

```
public class POIres implement Serializable{
private String gname;
private String category;
private double longitude;
private double latitude;
private String description;
private String address;
private String instruction;
private String file_path;
private double average_mark; //记录一个 POI 的平均得分
private String recommender; //推荐者的用户名（多个拼接），按朋友搜索时使用
private String no_recommender; //不推荐者的用户名
private int recommender_num; //推荐者人数
private int no_recommender_num; //不推荐的人数
}
```

## 5.4 数据库表的详细设计

根据客户端系统存储需要, POI 模块需要以下几个表: 用户属性表、POI 信息存储表和 POI 显示表。其中, 用户属性表存放用户身份信息, 用于网络查询时

的身份验证；POI 信息存储表存放用户的 POI 信息。

### 用户属性表

用户属性表作为用户登录服务器的身份鉴别。包含以下字段：

User\_id 表示当前用户的身份 id，该 id 用于在与服务器进行交互时进行身份验证。

Name: 表示用户的名称

Password: 用户密码，登陆服务器时使用。

表 5-1 用户属性表

Table	user_tab				
Purpose	存放用户信息				
Primary Key	User_id				
Foreign key	NULL				
Index	NULL				
No	Column			NULL	Illumination
	Name	Type	Length		
1	User_id	integer	32	N	用户 ID
2	name	Text	50	N	用户名称
3	password	Number	20	Y	用户密码

### POI 信息存储表

该表用于存储从服务器查询到的 POI 信息。包含以下字段：

poi\_id: 每个 POI 编号，作为该表的主键，为自增长序列。

Category: POI 的分类名称

Name: POI 的名称

Lat/Lon: POI 的经纬度信息

Desc: POI 的描述信息，该信息一般为服务器 GIS 数据库提供的基本信息。

Pic\_name: 标示该 POI 的图片的名称。不是每个 POI 都有对应的图片，如果没有则该项为空。

Mark、commend: 用户的好友对该 POI 的打分与评论。

表 5-2 POI 信息存储表

Table	poi_search_tab1
Purpose	存放从服务器查询到的 POI 信息
Primary Key	poi_id
Foreign key	NULL
Index	NULL

No	Column			NULL	Illumination
	Name	Type	Length		
1	poi_id	integer	32	N	poi 编号
2	category	text	300	Y	Poi 分类名
3	name	text	50	N	Poi 名称
4	lat	real	32	N	纬度
5	lon	real	32	Y	经度
6	desc	text	300	Y	该 poi 的描述
7	Pic_name	text	50	Y	Poi 图片的名字
8	mark	real	32	Y	该 poi 的评分
9	commend	text	300	Y	关于该 poi 的评论信息
10	poi_idx	integer	32	N	Poi 在服务器中的编号

### POI 显示表 poi\_search\_tab2

该表用于存放在电子地图上显示的 POI 的信息。该表的结构与 poi 查询表 poi\_search\_tab1 相同。

表 5-3 POI 显示表

Table	poi_search_tab2				
Purpose	显示查询到的 POI 信息				
Primary Key	poi_id				
Foreign key	NULL				
Index	NULL				
No	Column			NULL	Illumination
	Name	Type	Length		
1	poi_id	integer	32	N	poi 编号
2	category	text	300	Y	Poi 分类名
3	name	text	50	N	Poi 名称
4	lat	real	32	N	纬度
5	lon	real	32	Y	经度
6	desc	text	300	Y	该 poi 的描述
7	Pic_name	text	50	Y	Poi 图片的名字
8	mark	real	32	Y	该 poi 的评分
9	commend	text	300	Y	关于该 poi 的评论信息

10	poi_idx	integer	32	N	Poi 在服务器中的编号
----	---------	---------	----	---	--------------

## 5.5 客户端系统主控逻辑

客户端系统采用消息循环机制,主控函数为 `gtk_main()`,当程序进入 `gtk_main` 函数后,等待事件的发生,一旦发生某个事件,相应的信号将产生。如果程序中定义了相应的消息处理函数,系统会自动进行调用。

以 POI 功能的启动控制为例,其主要控制逻辑状态转移图如图所示:

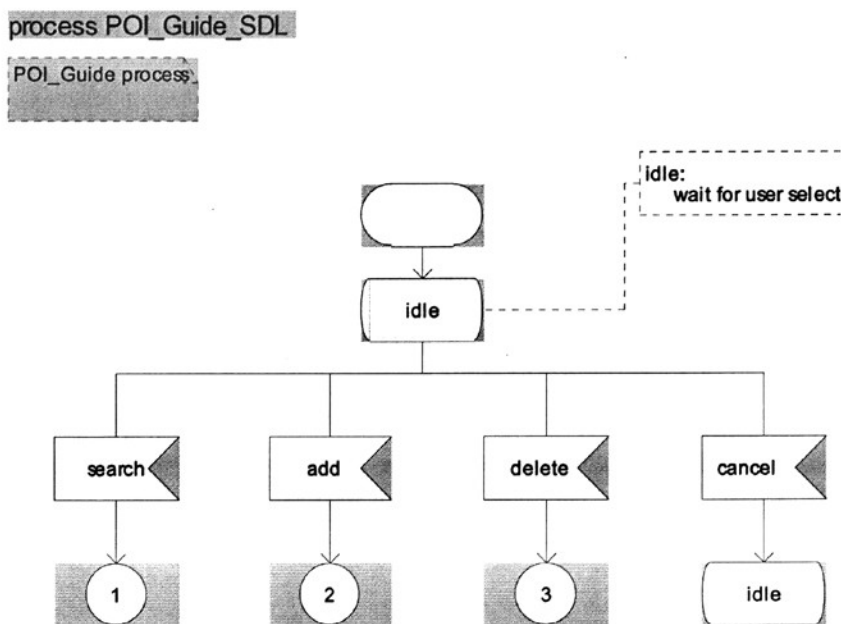


图 5-14 POI Guide 状态迁移

在选择 POI Guide 功能后, `gtk_main` 处于无限循环,系统进入等待用户选择状态 (`idle`),此时用户可以选择查询 (`search`)、添加 (`add`)、删除 (`delete`) 或者取消选项。根据用户的选择系统跳转到对应的状态中。

若用户选择 `cancel`,则重新回到 `idle` 状态。

当用户启动 POI 查询功能,程序进入 1 状态对应的回调函数,具体如下:

```
static gboolean
```

```
    poi_network_search_func (GtkAction *action)
```

启动网络查询 POI 系统会自动调用该函数,该函数将通过

`soap_call_ns1__QueryPOI()` 函数传送用户的查询请求,此时系统等待服务器的

查询结果，若查询成功，将结果数据保存到客户端系统数据库，并通过 `map_force_redraw()`和 `refresh_poi()`两个函数将查询到的 POI 所对应位置在电子地图上的位置显示出来。

若查询失败，系统会调用相应的错误处理函数告知用户。

```
static gboolean
```

```
    poi_local_search_func()
```

启动本地查询 POI，系统自动调用该函数在客户端数据库中寻找符合查询条件的 POI，然后调用 `map_force_redraw()`和 `refresh_poi()`刷新地图，将 POI 显示在电子地图上。

```
static gboolean
```

```
    upload_poi(char* poi_category, char* poi_label, double lat, double lon,  
char* poi_desc, char* pic_file)
```

该函数负责上传 POI 到服务器端。Pic\_file 为包含图片的 POI 的图片信息。该信息经过 base64 编码传送到服务器，服务器对其解码还原成原始图片。

```
static void
```

```
    poi_comment(GtkWidget * widget, int idx)
```

若用户对某个 POI 发表评论，通过该函数将用户的评论信息发送到服务器，该信息上传到服务器并经过认证后，对该用户的好友可见。



## 第六章 测试结果呈现及性能改进

本章主要对系统界面以及 POI 模块所实现的各个功能进行测试,包括系统界面的呈现、POI 查询、添加、删除、以及添加 POI 评论等功能的展示。

### 6.1 系统主界面呈现

下图为在缩放级为 4 级时程序主界面。屏幕主要区域为地图显示区域;工具条隐藏在地图的下方,右边为 4 个功能菜单,全屏时也可隐藏。

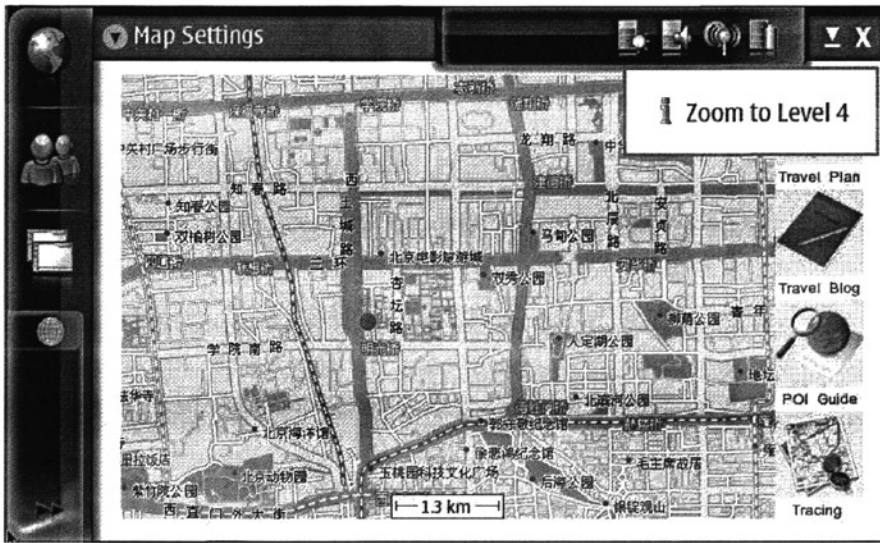


图 6-1 系统主界面

### 6.2 客户端系统 POI 功能展示

#### 6.2.1 POI 搜索功能

下面描述一个用户在移动终端的电子地图上查询某一类兴趣点的过程。首次使用该功能时需要登录服务器进行认证:



图 6-2 认证界面

服务器端通过用户登录可以获取到的该用户的基本信息：

- ✓ 用户当前位置
- ✓ 用户身份
- ✓ 社交网络信息：他的好友相关信息，包括有好友对该 POI 的评价，简单起见，定义对某兴趣点的评价为“推荐”与“反对”两种，将来若需要也可对该项进行扩展。

在主界面点击右侧 POI Guide 功能菜单，弹出 POI 功能按钮，如图：

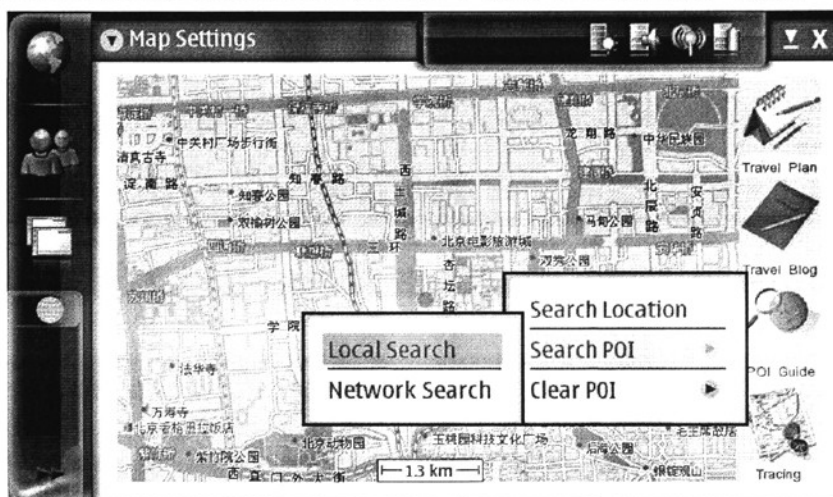


图 6-3 POI 查询界面

在 Search POI 子菜单里可以选择查询方式，这里选择网络查询，查询请求通

过 WLAN 传到服务器，再由服务器把查询结果返回。

用户在终端上输入查询条件：

- ✓ Category 表示欲查询兴趣点的分类，例如输入“School”表示查找学校；
- ✓ scale 表示查询范围，默认范围为 1.0km；
- ✓ 当查询你的好友推荐的 POI 时可以将 public 选项改为 friend，
- ✓ Top 5 POIs 表示在地图上显示 POI 的个数，图 6-4 表示只显示查询到的前 5 个 POI 数据。

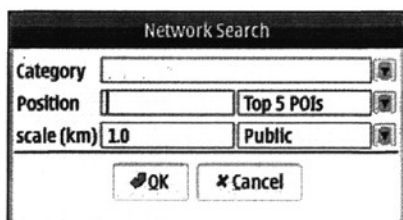


图 6-4 POI 网络查询输入框

该次搜索表示该用户想要搜索周围的学校，由于探测到该用户属性为大学生，所以搜索到的结果中大学校园放在靠前的位置，他的同学中如果有对学校的评价则显示在地图查询结果的描述区域中。

查询出的结果如图 6-5 所示：

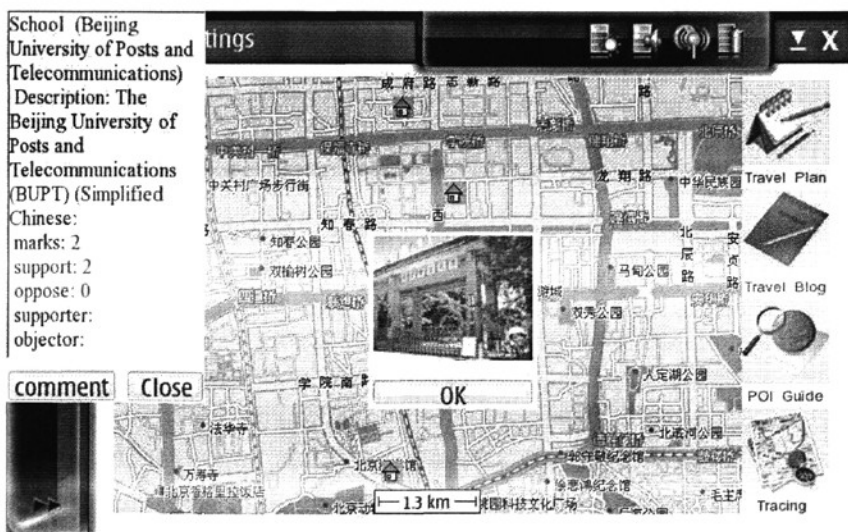


图 6-5 查询结果显示

可以看出,在该 POI 的描述中, marks 表明了用户(主要是用户的好友)对该地点的打分情况,这里取 5 分制,也就是说,“北京邮电大学”该 POI 的加权平均分为 2 分。Support 表示对该地点持好评态度的用户个数;oppose 表示对该地方持批评态度的用户个数。这样,对一个 POI 的描述就显得更加客观、全面。

如果用户自身对该 POI 有自己的看法,也可以通过 comment 选项添加自己的评论,这样其他人也可以看到该用户对该 POI 的评价。

对于呈现在地图上的不同分类的 POI,以不同的图标进行显示,这样,可以清楚的区分出地图上的不同种类的 POI。如图 6-6,图中紫色圆柱形图标代表商业建筑类别的 POI,而图 6-5 中的分类为“学校”的 POI 分类则以绿色房子作为标记。



图 6-6 POI 显示

## 6.2.2 POI 添加功能

如果用户发现自己感兴趣的地方并且想要和好友分享,便可以使用 POI 添加功能。只需在地图上点击该地点所在地,便可以弹出 POI 添加菜单,填写必要的信息后,就可以通过服务器进行上传。如下图:

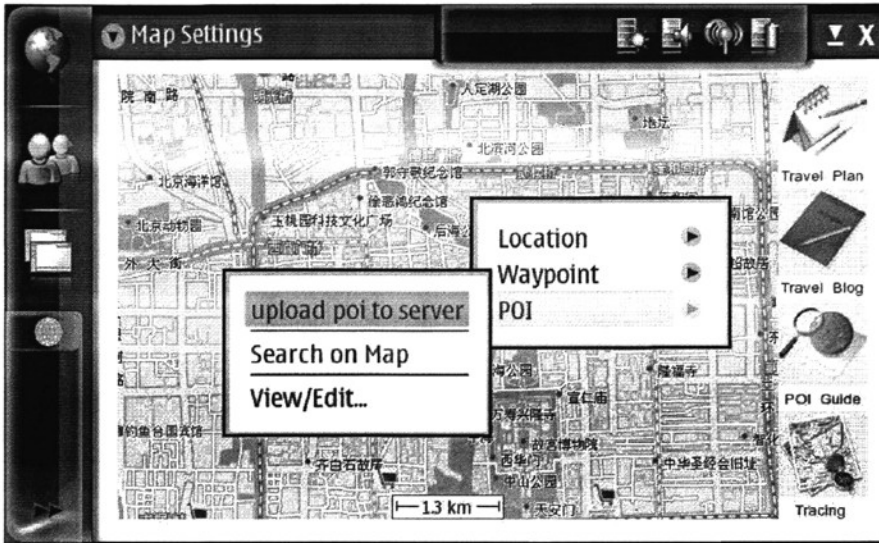


图 6-7 POI 添加功能

上传过程中，也可以传送该点的图片信息，只需要将该图片保存在客户端存储目录中，然后选中即可，如图：

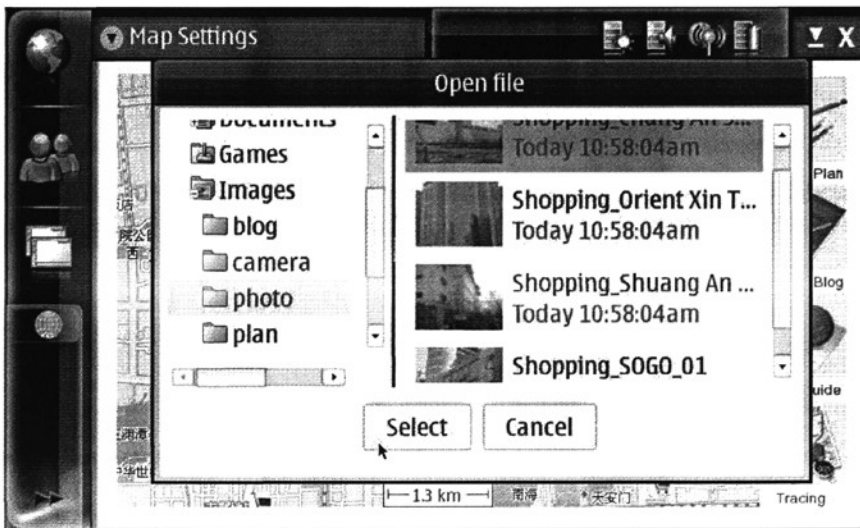


图 6-8 上传图片

### 6.2.3 POI 删除功能

该功能允许用户删除客户端存储的 POI 信息以及地图上显示的 POI。通过点击主界面右侧菜单即可启动该功能，如图 6-9：



图 6-9 POI 删除界面

从图 5-7 可以看出，POI 删除有两种情况：

(1) **Clear From Map** 表示只清除地图上显示的 POI 数据，而不删除在客户端数据库中对应的 POI。该方法清除 POI 后，使用本地查询功能（Local Search）还可以将 POI 标示在地图上。

(2) **Clear Permanently**：永久删除 POI，将地图上的 POI 和客户端数据库中的对应数据一起删除。删除后客户端数据库中的 POI 将无法恢复。

删除也可以分为逐个 POI 删除和批量删除。批量删除指删除客户端所有的 POI 数据或者删除某一类的 POI 数据。

批量删除功能在 POI 功能菜单中，逐个删除 POI 则需要点击地图上欲删除的 POI，便可以出现删除该 POI 的菜单，如图 6-10 所示：

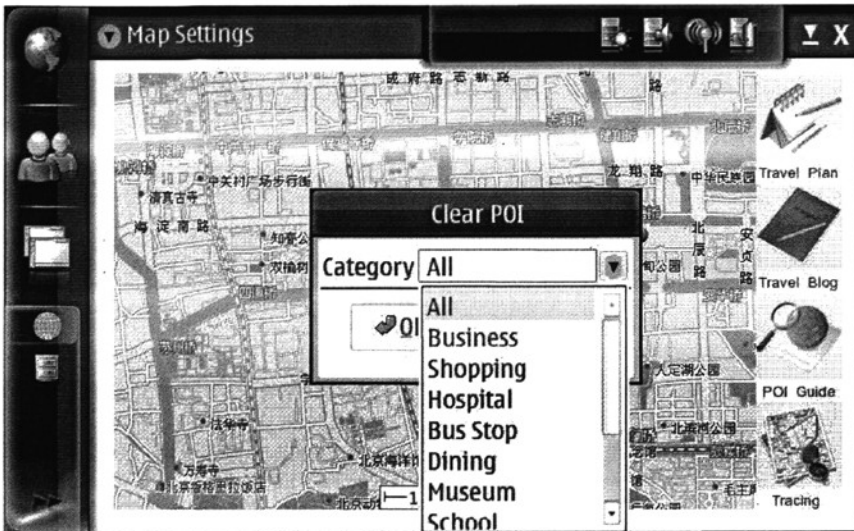


图 6-10 POI 删除分类列表

Category 列表列出了可以删除的 POI 的分类。选择“All”选项将删除所有分类的 POI 数据。

## 6.3 系统性能改进

### 6.3.1 数据库系统设计改进

由于客户端采用 N800 移动终端，属于资源受限系统，因此在该终端上不适合使用大型数据库系统，故客户端系统数据库采用 sqlite3 数据库。sqlite3 数据库普遍应用于嵌入式系统，该数据库为文件型数据库，数据存储方式默认为明文存储，所以像用户属性信息等一些含有用户隐私的敏感数据应当采用加密存储手段。然而，客户端系统为资源受限系统（N800 采用德州仪器 TI 2430 处理器，主频为 320Mhz），并且 sqlite3 数据库对于加密存储方式需要多消耗 40~50% 计算开销，因此，对一些记录庞大的非敏感数据来讲，则不宜采用加密存储方式。

综上，对于端系统数据库，不同的表应当采取不同的存储策略。下表为各个表单采用的存储策略：

表 7-1 数据库表存储方式

表	存储方式
user_tab	加密
poi_search_tab1	非加密
poi_search_tab2	非加密

ursite	加密
plan	非加密
site	非加密
category	非加密

### 6.3.2 超时系统

由于客户端与服务器之间的数据交互过程采用独立线程方式，并且超时时间均设定为固定值。例如，POI 网络查询超时时间设定在 8 秒。但是，无线网络应用环境相比有线网络更为复杂，网络的突发性和延迟性更加明显，因此，单一的超时设定在某些极端情况下便不太适合。那么，可以设计出根据网络负载状况自适应的调整超时或数据重传机制就显得尤为重要。

### 6.3.3 多级分类目录

当用户搜索某一类型的地点时，应设计多级分类目录来为其提供相关的结果。例如，当用户搜索“北京理工大学”时，会出现学校的位置及详细信息。而地图上还会显示一个该地址的分类目录，如设定此分类目录为“教育科研》学校》大学”。通过点击“大学”，会出现这个目录下的所有大学；点击“学校”，则出现该目录下的所有下级目录，例如小学，中学等等；同样的，点击“教育科研”，就会出现其下级目录的页面。所以说，通过这个多级分类目录，用户可以更方便的找到与所查找地点相关联的一些地点。



## 第七章 总结

### 7.1 论文总结

近些年来,随着计算机技术和空间信息技术的发展,电子地图获得了蓬勃发展。而随着网络的普及,网络电子地图也已经越来越广泛地被人们所利用。本文重点分析了移动环境下面向电子地图服务的客户端系统功能需求,提出了面向电子地图服务的客户端系统界面和用户兴趣点服务的设计,并实现了基于 Maemo 开发平台的客户端系统原型。并根据系统需求实现了可交互的移动电子地图服务。主要工作集中在如下几个方面:

1. 在对系统的需求分析的基础上,说明了面向移动电子地图服务的客户端程序的设计目标和原则,然后给出了用户兴趣点(POI)服务的总体设计,分析了其逻辑功能结构,在对客户端软件的功能进行进一步的细化,进行了模块化分解。并给出客户端系统数据库表的设计。
2. 论文介绍了客户端系统 POI 模块的详细设计,结合具体的开发环境,设计了易于开发的数据结构。分析了基于 GTK+应用程序的开发机制,设计了能够快速,并发处理消息的程序结构。给出主要模块的消息序列图和状态转移图,和数据库详细设计
3. 在前期完成上述工作的基础上,分析了现有的移动电子地图服务客户端系统的不足。提出了功能改进和性能优化方案,这种方案可以在传输效率、系统安全性以及功能特性上得到增强。

电子地图给人们提供了便利,可以让人们更方便的进行位置搜索、公交查询、线路导航等等。但该电子地图的功能不应该仅仅局限于这些,还需要更多的新功能使电子地图更智能化与人性化。所以作者结合了移动终端的特点和 web2.0 技术的思想,在电子地图中增加了许多交互性的服务,例如,用户兴趣点添加上传、好友评论等新功能,使其为不同的用户群提供更个性化的服务,能够更好的实现人机交互,也更方便于用户的使用。

从另一方面来看,该电子地图系统也存在一些效率与安全性的不足之处,另外某些流行的功能例如流媒体显示等还未能实现,这也是作者将来继续研究的工作之一。

### 7.2 研究生期间参与的主要工作

作者在研究生期间主要参与了两个项目的工作。第一个项目为航天部测试用

例生成与管理系统设计与管理，本人主要负责服务器数据库系统搭建与管理、设计系统信息交换字 XML 文件格式、服务器端单元测试与整体功能测试以及编码实现后台核心处理系统与前台的通信模块。

第二个项目为 NOKIA 移动环境下多维服务框架，论文作者主要负责设计客户端系统设计，功能模块划分，设计与实现电子地图上兴趣点 (POI) 模块功能以及撰写项目相应技术文档等。作者在移动环境下多维业务框架研究项目的工作中，前期完成了整个系统的结构设计，系统组件的划分与组合研究，然后在系统的体系结构与功能需求的基础上，重点研究并实现客户端系统的设计与功能实现，并在移动终端完成可交互的电子地图服务。

期间，完成如下文档：

- ✓ 测试用例管理系统信息关键字设计与说明文档
- ✓ 测试用例管理系统数据库表单设计文档
- ✓ 测试用例管理系统软件测试报告
- ✓ MMDSF 客户端系统需求分析说明书；
- ✓ MMDSF 客户端系统总体设计说明书；
- ✓ MMDSF 客户端系统详细设计说明书；
- ✓ MMDSF 客户端系统主界面设计说明书；

## 参考文献

- [1] John P Wilson, A Stewart Fotheringham, Pip Forer: Transaction in GIS, Oxford, Blackwell Publishers Limited, 2001.
- [2] CAI Zhongliang: Information Organization and Visualization Mechanism of Electronic Map, 地球空间信息科学学报, 2008年4期
- [3] M Yamamoto, K Ishikawa, K Ito: Electronic map display system, Ogawa, US Patent 4,737,916, 1988 - Google Patents
- [4] Maemo Development Platform: White paper<<http://www.maemo.org>>.
- [5] 哈洛.实用技术:开发Linux应用--用GTK+和GDK开发Linux图形用户界面应用, 电子工业出版社, 2000年, pp. 26-27
- [6] 周炎坤, 李满春, Web GIS开发方法比较研究, 计算机应用研究, 1999, pp.3-4
- [7] J de La Beaujardière, Web Map Service Implementation Specification, Open GIS Consortium, 2002.
- [8] Mrstik: Terrain Height Measurement Accuracy of Interface, metric Synthetic Aperture Radars, IEEE Transactions, 1996, GE-34(1): 219-228.
- [9] 钟耳顺, WebGIS--基于Internet的地理信息系统, 中国图象图形学报, A辑, 1998
- [10] Portable electronic map display - Patent D333463, April, 1988, Ogawa, 4780717
- [11] CD Wickens, P Kroft, M Yeh, Data Base Overlay in Electronic Map Design: Testing a Computational Model, ingentaconnect.com, 2000
- [12] J Grudin: Digital representation of context, 2004, pp.18-22
- [13] JH Hong, SY Lin: Web-based Thematic Map Service in OpenGIS, Environment - aars-acrs.org
- [14] Peter Seebach: 开发Nokia N800设置并测试构建环境, <http://www.ibm.com>, 2007年12月17日
- [15] Koperski K, Adihary J, Han J Ming: Knowledge in geographical data, communications of ACM 1999
- [16] Nan, LIU Ren-yi: Principle and Application of WebGIS, Beijing: Science Press, 2002
- [17] Luis Gravano, Hector Garcia-Molina, A. Tomasic: The Effectiveness of GIS for the Text-Database Discovery Problem, Proc. of the 1994 ACM SIGMOD, International Conference On Management Of Data, 1994.
- [18] <http://www.tianmomo.com/GIS/Knowledge/2005-8/99.html>

[19] Tutorial: <http://www.maemo.org/platform/docs/tutorials>

[20] Altomare, G Cascarano, C Giacovazzo, D Viterbo: E-map improvement in direct procedures, Foundations of Crystallography, 1991

## 附录一 缩略语

缩略语	英文全称	中文解释
<b>3G</b>	3 <sup>rd</sup> Generation	第三代通信技术
<b>E-map</b>	Electronic Map	电子地图
<b>CPU</b>	Central Processing Unit	中央处理器
<b>GDK</b>	Graphics Drawing Kit	绘图工具集
<b>GIS</b>	Geographic Information System	地理信息系统
<b>GPS</b>	Global Positioning System	全球定位系统
<b>GTK+</b>	GIMP Tool Kit	GIMP 工具集
<b>HTTP</b>	Hypertext Transfer Protocol	超文本传输协议
<b>PDA</b>	Personal Digital Assistant	个人数码助理
<b>POI</b>	Point Of Interest	兴趣点
<b>SNS</b>	Social Network Service	社会化网络服务
<b>SOAP</b>	Simple Object Access Protocol	简单对象访问协议
<b>TCP</b>	Transmission Control Protocol	传输控制协议
<b>UI</b>	User Interface	用户界面
<b>WLAN</b>	Wireless Local Area Network	无线局域网
<b>XML</b>	Extensible Markup Language	可扩展标记语言

## 致 谢

两年多的硕士学习生活即将结束。回首在宽带网中心这两年半的研究生学习和生活。不禁感慨万千。正是因为有许多人诚心的帮助、支持和鼓励,才使我取得能够受益终身的进步。饮水思源,借此良机,对所有关心和支持过我的良师益友和亲人们致意最衷心的感谢。

在论文完成之际,首先我要衷心感谢我的导师李玉宏老师。李老师在学习中给予我孜孜不倦的教诲,在生活上给予我亲切的关怀。她对研究工作的热情和严谨的治学态度给我留下了深刻的印象,对我未来的工作和学习将产生深远的影响。

感谢王文东老师和龚向阳老师的指导!从任务目标的制定,任务的分工到各功能的具体实现,老师们一直在给我提供各种宝贵的意见,并教会了我们如何学习一些相关新技术,并且如何才能尽快上手,将其应用到项目中,以及遇到技术难题时如何寻找解决途径最为高效,培养了我们独立思考,独立解决具体问题的能力。同时,对于每位同学的工作进度老师都十分关注,适时的监督和催促,我们每周的项目例会都按时进行,我们向老师汇报项目进展,遇到的困难等,老师给我们的情况进行分析,做出响应调整。再次对以上两位老师表示衷心的感谢!

衷心感谢实验室其它老师给予我的热情帮助。他们坚持创新、对学科前沿敏锐的洞察力以及做人的态度给我留下了深刻的印象。这些都将成为我珍贵的精神财富。

我还要感谢项目组里的蔡雅莉、欧锐、王晴、夏双荣、刘晓娜、秦灵伶和陈卫晓同学,以及实验室里的其他师兄师姐和师弟师妹们。这是一个充满活力和创造力的团体,身为这个团体中的一员,让我受益匪浅。

最后,感谢我身边所有一直关心我、帮助我、鼓励我的朋友们!谢谢你们!

## 攻读学位期间发表的学术论文

1. 李彬, “基于用户环境感知的电子地图搜索服务”, 中国科技论文在线, 2008年11月。