



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： 杜娟 日期： 2010.1.10

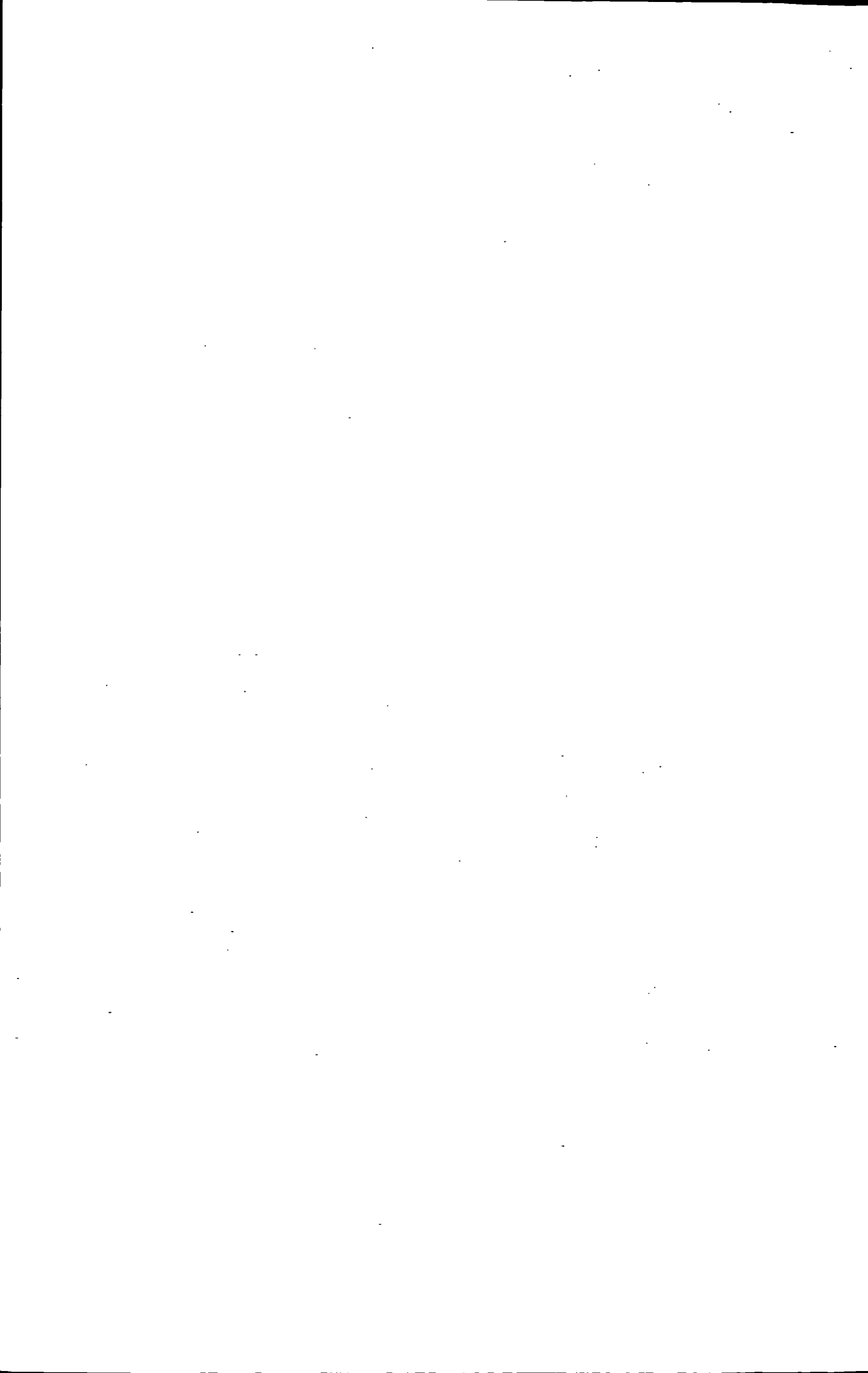
关于论文使用授权的说明

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： 杜娟 日期： 2010.1.10

导师签名： 郑志刚 日期： 2010.3.14



飞信机器人系统的设计与实现

摘要

IM (Instant Message) 巨大的用户资源, 良好的扩展性以及 IM 用户对 IM 的粘性使其成为互联网服务最有利的推广平台之一, 关于 IM 的应用成为研究的热点之一。IM 刚出现时就有人提出 IM 机器人能够成为 IM 杀手级的扩展应用。IM 机器人为一自动化的 IM 虚拟账号, IM 用户可以任意与之对话、查询信息。凭借强大的后台, IM 机器人可以为 IM 用户提供多种信息服务。2006 年中国移动推出的飞信是国内唯一能够跨平台使用的 IM, 短短三年时间, 用户渗透率就占据市场第二的位置。然而, 从成长阶段来说, 飞信还处在幼年期, 没有较多的扩展应用来实现增值服务, 飞信机器人能够借助飞信的优势, 实现无缝通信, 同时响应来自 PC 终端或手机终端飞信用户的信息服务请求, 是飞信比较有前景的一个扩展应用。

本文在参阅了大量 IM 机器人相关文献, 并且调研了如 QQ 机器人, MSN 机器人等的实际应用情况基础上, 设计与实现了飞信机器人系统, 该系统实现绝大部分飞信客户端功能, 使飞信机器人系统可以与飞信用户进行信息交互, 同时该系统实现简单、易用的二次开发接口, 开发人员能够在不改变系统架构的基础上, 方便快捷的将各种新兴的互联网应用集成到该系统中。通过与飞信机器人的对话, 飞信用户可以方便的获得信息服务。本文主要完成的工作如下:

- 1.对飞信机器人系统进行详尽的需求分析, 确定了系统实现的目标。
- 2.搭建协议测试环境, 解析出基于飞信协议的飞信通讯过程, 实现绝大部分飞信客户端功能。
- 3.采用 SOA 的架构, 结合插件, 多线程, Web Service 等具体技术设计系统架构, 使其具有良好的扩展性。
- 4.在 Visual studio 2005 编译环境下, 用 C#实现了飞信机器人系统的编码设计。

该系统在设计完成后，应用在上海、江苏、陕西等省市移动的官方机器人上，为飞信用户提供新闻，天气预报，笑话等信息服务。在移动的宣传下，利用飞信机器人进行了答题送话费等一系列优惠活动。在采取了很多吸引飞信用户的措施后，飞信机器人已经积累了很多活跃用户，奠定了用户基础。目前有很多服务提供商也在寻求和飞信机器人合作的方式，有一些增值业务即将被加入到飞信机器人系统中，飞信机器人作为载体，已经进入了向信息综合服务平台方向发展的轨道。

本文的主要章节安排如下：

第一章主要介绍课题研究背景与意义，解决的主要问题及系统实现的目标和效果。

第二章对系统的整体需求进行了全面的分析，并分析了技术可行性。

第三章在系统需求分析的基础上，进行系统设计。

第四章在系统设计的基础上，对系统实现的关键技术进行描述。

第五章在对本文进行总结的基础上，提出继续研究的方向和内容。

关键词：IM IM 机器人 飞信机器人 飞信协议 信息综合服务平台

Design and Implementation of Fetion Robot System

ABSTRACT

IM has a huge user resource, good scalability, and the viscosity of IM users, which makes it most beneficial to the promotion of Internet services. Now Extension applications of IM become one of the hot researches. IM robot is considered as important application when IM come into being. IM robot is an automated IM virtual account number and can provide lots of information services for IM user from which IM users can get information they interest in. In the year of 2006, china mobile launched Fetion which is the only IM can be used on cross-platform. Now Fetion has the secondly user number in domestic. However, Fetion is still in its infancy, there is little extended application to implement value-added services. Fetion robot is a promising Fetion application because it can satisfy information service request of both pc user and mobile user by the advantages of seamless communications.

After referring to a large number of IM robotics related literature, and researching in the practical application of QQ robots, MSN robots and so on, This paper design and implement Fetion robot system which achieve most of Fetion client-side function and has simple, easy interface to the secondary development. The Fetion user can talk with Fetion robot to get information they interest in, and developer can integrated a variety of emerging Internet applications without changing the system architecture. This paper completes the following tasks:

1. Perform detailed needs analysis to determine the system goal to achieve.
2. Build protocol test environment, analyze the Fetion communication process to achieve most of the Fetion client-side

function.

3. Design system architecture by using SOA architecture, combined with plug-in, multi-threading, Web Service and other specific technical, it has a good scalability.

4. Implements Fetion robot system by C# language under visual studio 2005 build environment.

The system design has been completed and used for the official Fetion robot in Shanghai, Jiangsu, Shanxi and other provinces and cities. Fetion robots provide news, weather, jokes and other information services. Besides, Mobile operator takes a lot of measures to attract user such as by using Fetion robot to answer question will get award and so on. Now Fetion robot have lots of active user. Several service providers are seeking the way to cooperate with Fetion robot. Several value-added services will be added to the Fetion robot system. Fetion Robot is developing toward the direction of integrated information service platform as a carrier.

The main sections of this paper are as follows:

The first chapter introduces the background and significance of the research, major problems to solve, objectives and result.

The second chapter performs full analysis of the system and analyses technical feasibility.

The third chapter performs system design based on system requirements.

The fourth chapter describes key technologies for system implementation based on system design.

The fifth chapter sums up the work of paper and proposes the direction of next research content.

KEY WORDS: IM IM Robot Fetion Robot Fetion protocol
Information integrated service platform

目 录

第一章 绪论	1
1.1 课题研究背景	1
1.1.1 IM 的发展现状	1
1.1.2 主流 IM 机器人的应用	1
1.1.3 飞信机器人	2
1.2 论文的主要工作及成果	3
1.3 论文的组织结构	4
第二章 飞信机器人系统需求分析	6
2.1 系统需求概述	6
2.1.1 系统网络结构图	6
2.1.2 通讯子系统需求	7
2.1.3 业务子系统需求	8
2.1.4 性能需求	9
2.2 技术可行性分析	9
2.2.1 .NET 技术	10
2.2.2 系统开发语言与开发环境	14
2.2.3 Socket 网络编程技术	15
2.3 本章小结	17
第三章 飞信机器人系统设计	18
3.1 SOA 技术简介	18
3.1.1 Web Service	19
3.1.2 .NET 对 SOA 的支持	21
3.2 系统总体架构	22
3.1.1 系统物理架构	23
3.1.2 系统逻辑结构	24
3.1.3 系统的运行机制	26
3.3 通讯子系统架构	27
3.3.1 通讯子系统功能结构	27
3.3.2 通讯子系统处理时序图	28
3.4 业务子系统架构	29
3.4.1 业务子系统逻辑结构	29

3.4.2 业务子系统功能结构	30
3.4.3 业务子系统处理时序图	31
3.5 本章小结	31
第四章 系统实现与应用	32
4.1 通讯子系统的实现	32
4.1.1 SIP-C 简要介绍	32
4.1.2 飞信的基本传输协议及通信过程	35
4.1.3 飞信客户端功能的实现	40
4.2 业务子系统的实现	43
4.2.1 业务流程	44
4.2.2 二次开发接口的实现	45
4.2.3 智能对话的实现	46
4.3 系统实际运行情况	48
4.4 本章小结	49
第五章 论文总结	50
参考文献	52
致 谢	54
攻读学位期间发表的论文	55

第一章 绪论

1.1 课题研究背景

近几年,即时通信(Instant Message,简称IM)已成为网民最喜爱的互联网应用之一,而IM的一项扩展应用IM机器人,以其便捷、智能的优点深得广大用户的喜爱,也成为互联网的研究热点。

1.1.1 IM的发展现状

据中国互联网络信息中心(CNNIC)在京发布的《第24次中国互联网络发展状况统计报告》^[1]显示,截止2009年6月30日,我国IM用户的数量已达到2.44亿,约占网民数量的72.2%。而据2009年底发布的《2009年中国即时通信用户调研报告》^[2]显示,截止2009年底,我国IM用户规模已经突破2.77亿。IM日益成为网民必不可少的通讯工具,绝大多数的网民会在上网的时候开启一种以上的即时通讯工具,人们已习惯于在IM上与亲人,好友,同事沟通。

IM巨大的用户资源,扩展性以及IM用户对IM的粘性使其成为互联网服务最有利的推广平台之一,IM的应用价值正在不断的被挖掘。服务提供商,涉及不同行业的企业都希望借助IM平台,获得庞大的用户群。即时通信软件的功能日益丰富,除了基本通讯功能外,逐渐集成了电子邮件、博客、音乐、电视、游戏和搜索等多种功能,IM的扩展应用成为互联网研究的热点方向之一。

1.1.2 主流IM机器人的应用

IM机器人是IM刚兴起时就提出的一种IM的扩展应用^[3-11]。它使用IM作为接口,以IM联系人的身份出现,能够根据IM用户的需求,通过与IM用户的会话,来为他们提供智能信息服务。企业可以利用IM机器人来代替人工客服人员,这样可以节约人力成本,其强大的学习能力以及24小时不间断的服务使得IM机器人甚至能够做得比人工客服人员更加出色;服务提供商可以将各种互联网应用集成到IM机器人上,借助IM机器人来为IM用户提供各类信息服务。IM机器人拥有强大的数据库、人工智能,易于扩展并且兼备即时通信软件所具有的良好特性,有能力成为联系IM用户和各类服务提供者的纽带。信息服务提供者可以将各种服务集成到IM机器人上,而IM用户可以通过IM机

器人人性的帮助来获取服务。IM 机器人的最终目标是成为信息综合服务平台。

主流的即时通讯厂商目前基本都有自己的 IM 机器人,如 QQ 拥有 QQLive、小 Q, MSN 拥有智能 IM 机器人小 I 等。QQ 机器人小 Q 是一个具有人工智能的聊天机器人,可以和 QQ 用户进行有趣的对话,得到 QQ 用户的喜爱。QQLive 机器人则作为 QQLive 的客服,提供一个 QQLive 的视频接口,QQ 用户可以在与其对话的即时消息框中通过链接直接连接到 QQLive 高清直播 web 界面。MSN 机器人小 i 具有更强大的功能,不止具备聊天的功能,还集成了诸如智能聊天,娱乐服务,生活服务等等很多应用。

IM 机器人的发展一定程度上受限于所归属的 IM。对于 QQ 来说,经历多年的发展,已经集成了多种增值业务,QQ 机器人作为众多扩展服务的一小种,没有受到特别的关注,尽管 2004 年时就有了第一个 QQ 机器人,但是到目前为止,并没有较多的功能扩展。对于 MSN 来说,它的用户集中在办公室白领人群,上班时习惯于用 MSN 进行沟通,借助 MSN 机器人可以更加方便快捷的获得需要的服务,MSN 机器人(例如小 I)得到了 MSN 用户的喜爱,根据 MSN 用户日益增长的需求,小 I 正在集成越来越多的服务,逐渐向提供综合服务方向发展。

1.1.3 飞信机器人

MSN 机器人的发展方向值得借鉴,但是受限于 MSN 不能实现跨平台应用的缺陷,MSN 机器人的信息传送平台局限在互联网以及电脑客户端,不可能同时满足 PC 端,手机用户的信息服务需求。2006 年中国移动借助强大的网络平台和资源优势,推出了飞信 IM,它是国内唯一能够跨平台使用的即时通讯工具,通过将“飞信”和手机号紧密绑定在一起,能够有效地防止网络欺诈,“飞信”客户端(PC 或手机)可以向手机发送短信,手机也可以下行发短信到飞信客户端,在即时通讯之间架起了无障碍使用的桥梁。超过 3 亿的移动用户都是飞信的潜在用户,据《2009 年中国即时通信用户调研报告》显示,移动飞信在 3 年左右的时间里用户渗透率已经高达 20.5%,占据市场第二的位置,已经超过了 MSN。

飞信^[12]能够迅速的获得如此大规模客户群体主要因为中国移动许诺了飞信业务永远免费向用户开放,“飞信”从 PC 客户端向手机、从手机客户端向外发送的短消息均为免费,用户实实在在感受到了实惠。另外飞信客户端做的也比较人性化,且比较稳定,让用户享受到了无缝通信的便利。然而,从成长阶段来说,飞信还处在幼年期,没有较多的扩展应用来实现增值服务,在未来的发

展模式上还要不断的探索。

飞信的定位不仅仅是初级 IM, 它更被视为 3G 时代的一大“金矿”。“飞信”体现了纵向整合能力, 真正实现了互联网和电信网络的统一; 其次, 飞信还体现了横向的整合能力, 在即时通讯之间架起了无障碍使用的桥梁。它标志着电信网同互联网的融合将向纵深方向发展, 多媒体数据业务成为未来电信业务发展的必然方向, 并将从根本上改变传统电信业务以语音为主的特征, 代之以融合话音、数据、图像等多种内容的传输业务。因此, 3G 时代运营商的角色也必将面临一个根本性的转变。电信运营商要适应新条件下的市场需求, 就必须提供能够横跨电信网和互联网的融合服务。从某种意义上讲, “飞信”业务的拓展不仅关系到中国移动的未来, 而且极有可能成为电信网、互联网融合的里程碑, 成为 3G 时代的移动门户, 乃至重要应用平台。

飞信机器人能够借助飞信的优势, 同时满足飞信用户在手机端和 PC 端的信息服务需求, 具备良好的扩展能力, 是飞信比较有前景的一个扩展应用。目前中国移动正在大力发展飞信的扩展服务, 能够给予飞信机器人足够的发展空间, 使飞信机器人有机会成为综合信息服务平台, 实现在任何时候, 任何地点以人性化的方式为飞信用户提供各类信息服务的目的。

随着飞信的兴起, 出现了一些用于系统监控的飞信机器人, 远程控制的飞信机器人, 提供群发短信功能的飞信机器人……。这些飞信机器人只适用于特定的应用, 处于初期的发展阶段。

1.2 论文的主要工作及成果

目前的飞信机器人只适用于特定的应用, 可扩展性不强。本课题旨在设计与实现一个飞信机器人系统, 底层通讯实现飞信客户端功能, 使飞信机器人系统可以与飞信用户进行信息交互, 同时该系统实现简单、易用的插件接口, 开发人员能够通过重写插件的方式, 方便快捷的将各种新兴的互联网应用集成到该系统中。

需要解决的主要问题有:

1. 飞信协议是中国移动自定义的 IM 协议, 协议细节没有公布, 怎样搭建协议测试环境来分析飞信协议是首先需要解决的问题。
2. 系统通讯子系统和业务子系统的的需求。
3. 如何采用智能客户端架构, 结合插件, 多线程, Web Service 等具体技术设计系统架构, 使其具有良好的扩展性。
4. 在特定的编译环境下, 将系统用代码实现。
5. 将系统应用到实际中。

目前已经成功的解决了上述 5 个问题,系统的通讯子系统和业务子系统相互独立,通讯子系统实现飞信机器人注册,登陆,获取好友列表,发送,接收消息等飞信 IM 客户端大部分功能。通讯子系统和业务子系统是独立运行的两个应用程序,之间通过命名管道通信。业务子系统使用功能菜单作为逻辑基础,根据飞信用户所处的状态定位到相应的插件进行处理。业务子系统对二次开发中可能需要的信息进行了最大化的抽象和提取,使二次开发中能将全部注意力集中到新应用的开发上。

尽管飞信已经得到了广大飞信用户的支持,但是飞信机器人几乎不被飞信用户所知,如果飞信机器人没有得到用户的粘性,服务提供商也不会对飞信机器人发生兴趣。要使飞信机器人按照设想的模式发展成为一个信息综合服务平台,必须按照以下三步进行:

- 1.飞信机器人如果要得到用户的粘性,首先得保证飞信用户信任飞信机器人,用户不会相信一个陌生的飞信号,所以,最好是移动来推出自己的飞信机器人,并且给予宣传。

- 2.使飞信用户对飞信机器人产生兴趣,将飞信机器人添加到好友联系列表里。这需要飞信机器人能提供信息类的服务(比如用户最关心的新闻,天气预报,游戏等),或者移动可以利用飞信机器人推出一些促销活动,让用户通过飞信机器人参加促销活动,提高飞信机器人的知名度,增加用户的粘性。

- 3.当官方的飞信机器人已经吸引了大量用户的时候,服务提供商可以和飞信机器人合作,将业务集成到飞信机器人系统中,采用方便的二次开发接口,可以方便快捷的实现。当飞信用户和服务提供商都被飞信机器人吸引时,就会进入好的发展模式,飞信用户提出需求,服务提供商满足需求,飞信机器人作为载体,就会加速向信息综合服务平台方向发展。

本课题对以上三步作出了尝试,和上海,江苏等省市移动进行合作,将飞信机器人系统应用到官方飞信机器人上,在二次开发过程中,将新闻,天气预报,游戏,彩铃,定位交友等几十余种信息服务集成到飞信机器人系统中,移动也利用飞信机器人推出了很多优惠活动,大大提高了飞信用户对飞信机器人的粘性。飞信机器人作为载体,已经进入了向信息综合服务平台方向发展的轨道。

1.3 论文的组织结构

根据论文的研究内容,主要的章节安排如下:

第一章 绪论主要介绍课题研究背景与意义,解决的主要问题及系统实现的目标和效果。

第二章 系统需求分析。对系统的整体需求进行了全面的分析，同时对本文所涉及的关键技术和技术可行性进行了说明。

第三章 飞信机器人系统设计。在系统需求分析的基础上，进行系统设计。

第四章 系统实现与应用。在系统设计的基础上，对系统实现的关键技术进行描述。

第五章 在对本文进行总结的基础上，提出继续研究的方向和内容。

第二章 飞信机器人系统需求分析

飞信机器人系统要实现飞信客户端的大部分功能和易于扩展的二次开发接口,按照功能分为通讯子系统和业务子系统两部分实现。本章综合考虑系统的性能和扩展性等因素,对系统进行详细需求分析,并进行技术可行性分析。

2.1 系统需求概述

本课题的目标是实现一个能够为大规模飞信用户提供服务的信息综合服务平台,该系统通讯接口实现飞信协议,能够无缝接入飞信通讯平台。

通讯子系统需实现飞信客户端的大部分基本功能,可以使用飞信账号登录、注销、变更在线状态、加入好友、向好友发送短信或者向好友飞信 PC 客户端发送信息等,通讯子系统需对系统底层通讯及协议进行完整的封装,业务子系统只需处理具体的业务不需要考虑通讯过程。

2.1.1 系统网络结构图

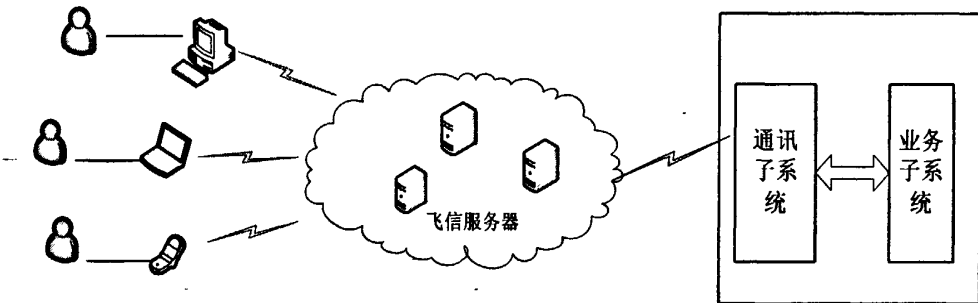


图 2-1 系统网络结构图

如图 2-1 所示,飞信机器人和飞信用户之间的通信以飞信为载体。飞信用户添加飞信机器人的账号为好友,或者飞信机器人主动添加飞信用户,来实现与飞信机器人的直接通讯,通讯子系统收到飞信客户发来的协议信息后,将协议报头脱去,再发送给业务子系统,业务子系统根据飞信用户当前的状态以及飞信客户发送的信息选择相应的业务处理策略进行处理,将返回的信息发送给通讯子系统,通讯子系统再打上协议报头,按照底层的通讯规则将协议数据报发送给飞信用户。

2.1.2 通讯子系统需求

通讯子系统主要负责与飞信服务器的信息交互以及与业务子系统的信息交互。

2.1.2.1 与飞信服务器的信息交互需求

需实现飞信客户端的大部分功能，并且需考虑系统容错情况，详细需求如下：

1. 注册，登录，注销功能。

用正式的飞信账号向飞信服务器注册，验证身份后，登录到飞信服务器，在关闭应用程序时，需向飞信服务器注销。

2. 主动回应好友添加请求，自动将好友资料入库。

3. 获取飞信联系人列表，将飞信联系人的更新信息存放在数据库中。

4. 向飞信好友发送文本信息，接收飞信好友发送的文本信息。

5. 当与飞信服务器连接意外中断后，提供数据缓存功能，能够自动尝试重新建立连接。

6. 具有流量检测控制的功能，在检测到流量过大时，主动断开与业务子系统的连接，自动回复飞信用户“系统忙，稍后再试”，避免将流量再引入业务子系统，造成系统崩溃。等到流量下降到系统可以承受时，再重新建立与业务子系统的连接。

2.1.2.2 与业务子系统的信息交互需求

通讯子系统与业务子系统相互独立，通讯子系统需将飞信用户发送的数据去掉协议头部转发给业务子系统，然后等待接受业务子系统返回的信息，再将其封装发给飞信用户。考虑功能要求和系统性能要求两方面，详细需求如下：

- (1) 通讯子系统和业务子系统是两个独立的应用程序，但是他们之间要进行非常频繁的、文本的消息通信，为了保证系统的性能，必须采用可靠的通讯机制。

- (2) 通讯子系统不仅需把飞信用户发送的信息转发给业务子系统，而且需把用户的状态（如飞信用户是使用手机，或 PC 发送的消息，在线或隐身等）发送给业务子系统，使之能做出正确的处理。

- (3) 当用户并发流量非常大时，只让一个飞信账号承担所有的流量

会导致系统崩溃，解决方法是让几个飞信账号作负载均衡，共同应对大的流量。这就要求业务子系统能够同时和多个通讯子系统通信。他们之间可以是主从的关系。

2.1.3 业务子系统需求

业务子系统需要提供简单的二次开发（SDK）接口，使服务提供者不需要考虑具体的飞信协议及其通信用程，只需要考虑具体的业务实现，把业务开发从繁杂的系统通讯开发中解脱出来。

业务子系统是整个飞信机器人系统的控制台，实现和飞信用户的智能信息交互，该业务平台提供的业务由插件提供，根据不同的插件配置方式，可以实现不同功能，比如在配置文件中配置天气预报插件，则程序运行时，自动加载天气预报插件，提供天气预报服务。如果在配置文件中同时配置天气预报和新闻插件，则能够同时提供天气预报和新闻服务。实现考虑如下情况：

(1) 接收通讯子系统发送过来的消息，经处理后将反馈消息发送给通讯子系统。

(2) 能够根据配置文件主动下行发送消息给通讯子系统，使之执行相应任务。

(3) 使用菜单的搜索方式：飞信机器人系统是一个提供综合服务的入口，如果说 Web 应用的 HTML 呈现需要用户点击和输入进行交互的话，飞信机器人的呈现则主要靠文字输入作为交互，为了使逻辑清晰，能够将多种不同服务有效组织起来，使用功能菜单作为逻辑基础，最终飞信用户可以通过两种方式直接进入相应菜单，第一种，回复相应菜单的序号进入下一级目录，这是一种基本的实现方式；第二种，飞信用户可以通过发送想要服务的关键字来直接进入相应菜单；第三种，人工智能方式，用户可以输入任何自然语言，强大的后台经过分析，进入相应的功能菜单级进行处理。当前的阶段是实现第一种和第二种，即实现系统的骨架，确保系统的稳定，可扩展能力。今后则考虑加强自然语言，人工智能方面的研究，施之以血肉，实现高智能的飞信机器人。

(4) 可以在配置文件中定义皮肤：飞信机器人系统是搭建在飞信之上的第三方服务提供者，尽管目前只实现文字输入作为交互方式，但是语言的组织方式，文字的字体，颜色，图片的不同都会给最终用户带来不一样的视觉和心理感受。考虑飞信机器人系统最终的使用者并不是开发人员而是不懂程序开发的策划人员，系统必须有简单易懂的配置界面，在本文设

计的飞信机器人系统中，需要固化几套皮肤，供配置时候选择。如果有特殊的需要，系统必须能够支持策划人员配置自定义的皮肤。

(5) 必须有详细充分的事件记录。程序自动运行在服务器上，看不见摸不着，如果不做好事件记录，一旦发生问题，将会带来不可避免的灾难后果。详细的事件记录也可以使管理员看到过去的时间里程序运行的大量细节，更好的监测程序的运行。

(6) 扩展性：当系统应用需要加载新的业务，去除老的业务时，只需要重写二次开发接口，不需要改变系统架构，在应用时，只需根据需要修改配置文件即可。

2.1.4 性能需求

飞信机器人系统的开发首先要求程序可靠，在一般情况下运行无差错，不发生内存泄漏之类影响程序运行的错误事件。但是如果遇见种种由于系统问题产生的错误，比如网络连接时间超时，系统负载过大等要有错误处理机制。所有偶然发生的错误事件，即使概率为 1/10000 也要能够考虑到。

系统要有良好的容错性，当用户进行非法操作或系统本身出现问题时要能正常退出程序，避免程序出现假死的情况。

系统要尽可能的高效，对系统的硬件条件要求尽可能低，运行时内存占用尽可能小，响应速度要尽可能快。

系统需要容易维护，应该采用模块化开发，各个模块之间不要有太多的联系，以免维护困难。

把飞信机器人系统的开发提高到“工程”的高度，预料到可能出现的问题，控制每个实践环节，防患于未然。软件从定义到开发、使用和维护，直到最终被弃用，要经历一个漫长的生存周期，需用一定的流程将各个环节串起来，并用规范的方法操作全过程。程序的设计需要遵循软件工程的理论和方法，提高软件的质量与生产率，在程序设计的过程中，要考虑到影响软件质量的各种因素，如正确性、性能、可靠性、容错性、易用性、灵活性、可扩充性、可理解性、可维护性等。只有设计出性能优良的飞信机器人系统，才能实现其成为信息综合服务平台的目标。

2.2 技术可行性分析

综合 2.1 节功能需求，本课题将基于 Windows 操作系统，采用 .NET

相关技术^[13-14]，Socket 网络编程技术^[15]等来搭建系统，在 visual studio 2005^[16]开发平台上，用 C#开发语言^[17-20]来完成系统的实现，下面将对相关技术原理做介绍，证明采用相关技术能够实现系统目标。

2.2.1 .NET 技术

系统设计的关键，主要是定义和说明包（子系统），以及包与包之间的相互依赖与通信机制。包通常所需要处理的要么是一个具体的功能区域（业务逻辑），要么是一个具体的技术区域（技术逻辑）。业务逻辑主要考虑的是对系统业务功能的实现，而技术逻辑则是进一步考虑用户界面、数据库或通信机制等形成的方案。把技术逻辑和业务逻辑区分开来是及其重要的，这是为了当修改程序的某一部分时不会对另一部分产生影响，更加便于进行“复用”，同时易于应对来自业务逻辑的变更需求。

.NET 拥有先进的软件体系结构的理念，.NET 开发人员能够将架构设计理念采用.NET 相关技术实现，采用.NET 相关技术进行软件设计，能够轻松创建出各种强大的应用程序。利用.NET 技术来进行飞信机器人系统架构设计，有能力将先进的理念应用到实际中。在第三章中，将应用.NET 技术来设计飞信机器人系统，下面对.NET 技术的核心，包括.NET 框架（CLR，.NET Framework 类库），ADO.NET，.NET 反射机制做简要介绍，这些在系统的设计中都起着至关重要的作用。

2.2.1.1 .NET 框架

.NET 开发平台中的.NET 框架是.NET 软件构造中的核心部分，.NET 框架为 Web 服务及一般 Windows 应用提供了执行环境，它集成了语言开发、代码编译、组件配置、程序运行和对象交互等各个层面的功能。在.NET 平台上创建的应用程序运行需要 CLR(Common Language Runtime)和.NET Framework 类库两个核心模块。

(1) CLR

CLR 是.NET 应用程序的托管代码执行环境，在组件的运行时报和开发时都起到很大的作用。在组件运行时，CLR 除了负责满足此组件在其他组件上可能具有的依赖项之外，还负责管理内存分配、启动和停止线程及进程，以及强制执行安全策略等。由于做了大量的自动处理工作（如内存管理等），CLR 使开发人员的操作非常简单。基于 CLR 开发的代码称为受控代码，它的运行步骤大体如下：首先使用一种 CLR 支持的编程语言编写源

代码, 然后使用针对 CLR 的编译器生成独立于机器的微软中间语言 (Microsoft Intermediate Language MSIL), 同时产生运行所需的元数据, 在代码运行时再使用即时编译器 (Just In Time Compiler, JIT) 生成相应的机器代码来执行。

(2) .NET Framework 类库

.NET Framework 类库是一组可重用的类集合, 这些类是面向对象的, 可以用于各种编程语言。该类库提供了几乎所有应用程序都需要的公共代码, 以及在此之上的应用程序模板, 这些模板为开发网络站点和网络服务提供各种特定的组件和服务, 包括从传统的命令程序、Windows 图形界面程序到面向分布式计算的 ASP.NET 和 Web 服务应用等。

.NET Framework 类库为程序员提供了大量的基础组件, 使程序员从繁重的编程细节中解放出来, 专注与程序的商业逻辑。它将核心 Win32 API 最常用的功能和外挂 SDK 的功能封装到一个统一的包中, 并对类库进行分组和描述, 使开发人员能够更方便地找到其应用程序所需的功能。

.NET Framework 类库通过名称空间组织, 它使用一种层次化的命名方法。在 .NET 框架中, 根或顶级名称空间是 "System", 在它之下按照功能区的分级制度进行排列。名称空间被存储在可移植运行模块文件中, 一般为 DLL 或 EXE 文件。 .NET Framework 的类通常位于相对应的名称空间中, 名称空间与名称空间及类之间通常用 "." 隔开。对于一个完整的类名, 类名中最后一个点号前的部分, 通常为名称空间, 而最后一个点号后的部分为类。通过命名规则将相关类归入名称空间的方法, 对于类库的建立和文档化是非常有好处的。一个名称空间可能由多个部件组成。而一个部件也可能包含多个名称空间中的类。

.NET Framework 有助于缩短产品开发时间, 简化发布和管理, 提高运行效率, 主要包括以下优点:

a. 可以使用任何编程语言: .NET Framework 允许开发者以任何语言进行开发, 使不同语言开发的程序结合得更紧密, 并使现有的开发技巧得以继续使用。

b. 减少了代码编写量: .NET Framework 使用高度模块化设计, 使得开发人员可以将更多精力集中到处理应用逻辑方面, 而不必再把时间花费在写 IDL (Interactive Data Language) 和 Register 代码上。

c. 以 XML/SOAP 为核心: .NET Framework 目标是将软件转化为服务, 因此采用基于 XML 和 SOAP 系列的集成标准。用户只需简单的注出所需的方法调用, .NET Framework 就能将它们转化为完整的 Web 服务。

d. 提高了应用程序的可靠性：.NET Framework 引入了新的技术使程序运行得更加可靠，比如以 .NET Framework 来管理内存、线程及进程，确保内存泄漏将不再发生。此外，ASP.NET 还监视 Web 程序的运行，并根据管理员设定的时间间隔，每过一段时间自动地重新执行一次这些程序。

e. ADO.NET 组件：ADO.NET 为基于网络并且可扩展的应用程序和服务提供数据访问的服务，它不仅支持传统的基于连接和游标的数据访问，而且更适合于无连接数据访问，并且支持 XML 及非关系模型的数据结构。

f. 标准化：微软已经将 .NET 框架的核心部分——公共语言架构(CLR 的一个子集，Common Language Infrastructure)提交给欧洲计算机制造商协会(ECMA)。现在正全力参与 ECMA 的标准化推广，并协助 ECMA 控制和管理这些标准。

2.2.1.2 ADO.NET

ADO.NET 是与 C#和 .NET Framework 一起使用的类集的名称，用于以关系型的、面向表的格式访问数据。这包括关系数据库，比如 Microsoft Access 和 SQL Server,以及其他数据库。数据共享使用者应用程序可以使用 ADO.NET 来连接到这些数据源，检索、操作和更新数据。

ADO.NET 的主要特点是引入了离线数据集(Disconnect DataSet)。所有的数据都可以离线使用而不必一直连接到数据库，只有需要对数据库进行编辑时才连接到数据库。对于 Web 应用程序而言，因为开启数据库联机在分布式的环境中会占用较多资源，所以离线数据存取可使应用程序具有高度的延展性。Dataset 是一个驻于内存的数据缓冲区，是一个分离的用于表示数据集合的独立实体，它提供了数据的关系型视图。不管数据来源于一个关系型数据库，还是一个 XML 文档，Dataset 都使用同样的程序模板和相同的内存区进行操作。它替代了原有的 Recordset 对象，提高了程序的交互性和可扩展性，因而非常适合于结构松散的、本质非连接的 Web 应用程序。

ADO.NET代表了处理位于本地或者分布式数据库中数据的下一个革新，具有如下的优势：

(1) 以往ADO的数据存取处理主要依赖于两层结构，而现在 ADO.NET的数据处理是采用三层以上结构，构建了一个更为合理的数据处理环境。

(2) ADO.NET是基于XML的，这表示几乎任何数据类型都可以被支

持和传递。

(3) DataSet使得ADO.NET摆脱了由ADO所提供的基于连接的模型,而将其转变成一个更加类似消息的、面向无连接的模型。这样遍历一DataSet可以不必打开到数据源的任何连接。这一改变带来了许多好处,包括通过防火墙发送数据的能力,消除了COM中的类型转换,以及一种真正脱机处理数据的方法。

2.2.1.3 .NET 反射机制

反射(Reflection)是.NET中的重要机制,通过反射,可以在运行时获得.NET中每一个类型(包括类、结构、委托、接口和枚举等)的成员,包括方法、属性、事件,以及构造函数等。还可以获得每个成员的名称、限定符和参数等。有了反射,即可对每一个类型了如指掌。如果获得了构造函数的信息,即可直接创建对象,即使这个对象的类型在编译时还不知道。

程序代码在编译后生成可执行的应用,首先要了解这种可执行应用程序的结构。应用程序结构分为应用程序域—程序集—模块—类型—成员几个层次,公共语言运行库加载器管理应用程序域,这种管理包括将每个程序集加载到相应的应用程序域以及控制每个程序集中类型层次结构的内存布局。

程序集包含模块,而模块包含类型,类型又包含成员,反射则提供了封装程序集、模块和类型的对象。我们可以使用反射动态地创建类型的实例,将类型绑定到现有对象或从现有对象中获取类型,然后调用类型的方法或访问其字段和属性。反射通常具有以下用途。

(1) 使用 Assembly 定义和加载程序集,加载在程序集清单中列出模块,以及从此程序集中查找类型并创建该类型的实例。

(2) 使用 Module 了解包含模块的程序集以及模块中的类等,还可以获取在模块上定义的所有全局方法或其他特定的非全局方法。

(3) 使用 ConstructorInfo 了解构造函数的名称、参数、访问修饰符(如 public 或 private)和实现详细信息(如 abstract 或 virtual)等。使用 Type 的 GetConstructors 或 GetConstructor 方法来调用特定的构造函数。

(4) 使用 MethodInfo 了解方法的名称、返回类型、参数、访问修饰符(如 public 或 private)和实现详细信息(如 abstract 或 virtual)等。使用 Type 的 GetMethods 或 GetMethod 方法来调用特定的方法。

(5) 使用 FieldInfo 了解字段的名称、访问修饰符(如 public 或 private)

和实现详细信息（如 `static`）等，并获取或设置字段值。

(6) 使用 `EventInfo` 了解事件的名称、事件处理程序数据类型、自定义属性、声明类型和反射类型等，添加或删除事件处理程序。

(7) 使用 `PropertyInfo` 了解属性的名称、数据类型、声明类型、反射类型和只读或可写状态等，获取或设置属性值。

(8) 使用 `ParameterInfo` 了解参数的名称、数据类型、是输入参数还是输出参数，以及参数在方法签名中的位置等。

`System.Reflection.Emit` 命名空间的类提供了一种特殊形式的反射，可以在运行时构造类型。

2.2.2 系统开发语言与开发环境

2.2.2.1 C#开发语言

.NET 技术可以帮助我们开发出稳定，高效的应用程序。.NET Framework 是.NET 的核心，如果想用到.NET 技术的各种特性来完成程序设计，就需要使用.NET 代码库编写代码（可以是任何一种语言）。

C#是 Microsoft 专门为使用.NET 平台而创建的。因为 C#是近期发展起来的，所以吸取了以前的教训，考虑了其他语言的许多优点，并解决了它们的问题。

使用 C#开发应用程序比使用 C++简单，因为其语法比较简单。但是，c#是一种强大的语言，在 C++中能完成的任务利用 C#也能完成。C#中与 C++比较高级的功能等价的功能（例如直接访问和处理系统内存），只能在标记为“不安全”的代码中使用。这个高级编程技术是非常危险的，因为它可能覆盖系统中重要的内存块，导致严重的后果。

C#只是.NET 开发的一种语言，但是能够认为它是实现.NET 技术最好的一种语言。因为它是唯一为.NET Framework 设计的语言，是在移植到其他操作系统上的.NET 版本中使用的主要语言。要使其他语言如 VB.NET 尽可能类似于其以前的语言，且仍遵循 CLR,就不能完全支持.NET 代码库的某些功能。但 C#能使用.NET Framework 代码库提供的每种功能。

2.2.2.2 Visual studio 2005

Visual Studio2005（以下简称 VS2005）是一种强大的集成开发环境，支持 C#(以及托管和非托管 C++、Visual Basic 和其他一些语言),这个环境

的优点是便于把.NET 功能集成到代码中。虽然可以使用基本的文本编辑器（例如 Notepad）来处理 C#源代码文件，再使用命令行应用程序（是.NET Framework 的一部分）把代码编译到程序集中。但是 VS2005 如下的优良特性使其当之无愧成为.NET 开发首选工具。

(1) 可以自动执行编译源代码的步骤，同时可以完全控制重写它们时应使用的任何选项。

(2) 文本编辑器可以配合 VS2005 支持的语言（包括 C#），这样就可以智能检测错误，在输入代码时给出合适的推荐代码。

(3) VS2005 包括 Windows Forms 和 Web Forms 设计器，允许 UI 元素的简单拖放设计。

(4) 在 C#中，许多类型的项目都可以用已有的“模板”代码来创建，不需要从头开始。各种代码文件通常已经为我们准备好了，减少了从头开始一个项目所花的时间。

(5) VS2005 包括几个可自动执行常用任务的向导，它们可以在已有的文件中添加合适的代码，而不需要考虑（在某些情况下）语法的正确性。

(6) VS2005 包含许多强大的工具，可以显示和导航项目中的元素，这些元素可以是 C#源文件代码，也可以是其他资源，例如位图图像或声音文件。

(7) 除了在 VS2005 中编写应用程序比较简单外，还可以创建部署项目，以易于为客户提供代码，并方便地安装该项目。

(8) 在开发项目时，VS2005 可以使用高级调试技巧，例如能一次调试一行指令，并监视应用程序的状态。

2.2.3 Socket 网络编程技术

飞信机器人通讯子系统与飞信服务器通信基于采用非标准 SIP，协议栈实现的是 TCP 承载。

Microsoft.Net Framework 为应用程序访问 Internet 提供了分层的、可扩展的以及受管辖的网络服务，其名字空间 System.Net 和 System.Net.Sockets 包含丰富的类可以开发多种网络应用程序。.Net 类采用的分层结构允许应用程序在不同的控制级别上访问网络，开发人员可以根据需要选择针对不同的级别编制程序，这些级别几乎囊括了 Internet 的所有需要--从 socket 套接字到普通的请求/响应，更重要的是，这种分层是可以扩展的，能够适应 Internet 不断扩展的需要。

抛开 ISO/OSI 模型的 7 层构架，单从 TCP/IP 模型上的逻辑层面上

看，.Net 类可以视为包含 3 个层次：请求/响应层、应用协议层、传输层。WebRequeust 和 WebResponse 代表了请求/响应层，支持 Http、Tcp 和 Udp 的类组成了应用协议层，而 Socket 类处于传输层。可以如图 2-2 示意：

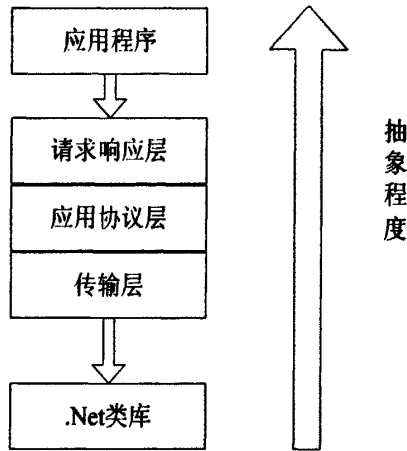


图 2-2 Socket 网络图

可见，传输层位于这个结构的最底层，当其上面的应用协议层和请求/响应层不能满足应用程序的特殊需要时，就需要使用这一层进行 Socket 套接字编程。

而在.Net 中，System.Net.Sockets 命名空间为需要严密控制网络访问的开发人员提供了 Windows Sockets (Winsock) 接口的托管实现。System.Net 命名空间中的所有其他网络访问类都建立在该套接字 Socket 实现之上，如 TCPClient、TCPListener 和 UDPClient 类封装有关创建到 Internet 的 TCP 和 UDP 连接的详细信息；NetworkStream 类则提供用于网络访问的基础数据流等，常见的许多 Internet 服务都可以见到 Socket 的踪影，如 Telnet、Http、Email、Echo 等，这些服务尽管通讯协议 Protocol 的定义不同，但是其基础的传输都是采用的 Socket。

其实，Socket 可以像流 Stream 一样被视为一个数据通道，这个通道架设在应用程序端（客户端）和远程服务器端之间，而后，数据的读取（接收）和写入（发送）均针对这个通道来进行。

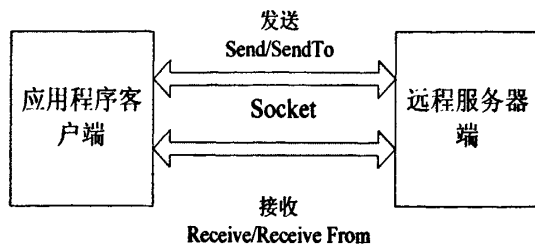


图 2-3 Socket 通讯机制

可见，在应用程序端或者服务器端创建了 Socket 对象之后，就可以使用 Send/SentTo 方法将数据发送到连接的 Socket，或者使用 Receive/ReceiveFrom 方法接收来自连接 Socket 的数据。

针对 Socket 编程，.NET 框架的 Socket 类是 Winsock32 API 提供的套接字服务的托管代码版本。其中为实现网络编程提供了大量的方法，大多数情况下，Socket 类方法只是将数据封送到它们的本机 Win32 副本中并处理任何必要的安全检查。

2.3 本章小结

本章确定了系统的实现目标，描述了系统的网络结构，将系统分为通讯子系统，业务子系统，网络通讯几个部分进行了详细的功能需求分析，最后对技术实现的可行性进行了分析，同时对系统涉及到的关键技术原理进行了说明。

第三章 飞信机器人系统设计

根据第二章的需求分析,飞信机器人系统由通讯子系统,业务子系统两部分组成。飞信机器人系统的目标是成为飞信的综合信息服务平台,对稳定性,实时性,可扩展性,可配置性的要求都很高。本课题采用了 SOA^[21-23]的体系结构,结合 Web Service,多线程,插件等具体技术来设计系统。设计过程采用 UML^[24](Unified Modeling Language)建模方式,这样能够专注于建立产品的模型和结构。

3.1 SOA 技术简介

随着时间推移,功能被说明,发布或使用的抽象级别会逐渐越来越高。我们已经经历了从模块化、面向对象到现在的面向服务的发展过程。SOA (Service-Oriented Architecture) 代表包含一系列当代技术的新一代架构平台。“面向服务架构”代表了一个模型,其中的自动化逻辑可分解为小的、不同的逻辑单元。这些单元组成一个较大的业务自动化逻辑片段,并可以对这些单元进行独立地分布式部署。SOA 鼓励单个逻辑单元自治而不相互孤立。逻辑单元仍要遵从允许其独立的一系列原则,同时充分维护其通用性和标准化。在 SOA 内部,这些逻辑单元就是所谓的“服务”。在基于 SOA 架构的系统中,具体应用程序的功能是由一些松耦合并且具有统一接口定义方式的组件(也就是 Service)组合构建起来的。

SOA 改变过去的技术组件为服务,强调的是技术无关性,关注的是实现怎样的业务功能——在业务请求与响应之间随时搭建快速通道,同时,变过去的紧耦合为松耦合,既保证系统弹性,又不失系统效率,进而实现重复利用软件资源、快速响应市场需求变化、提高生产力等目标。

SOA 帮助企业信息系统迁移到“leave-and-layer”架构之上,这意味着在不用对现有的企业系统做修改的前提下,系统可对外提供 Web 服务接口,这是因为它们已经被可以提供 Web 服务接口的应用层做了一层封装,所以在不用修改现有系统架构的情况下,SOA 可以将系统和应用迅速转换为服务。SOA 带给用户的好处很明显,可以降低开发成本,提高系统集成度和响应速度等,还能帮助解决因为系统升级带来的烦恼。传统的软件升级对用户就意味着每三年来一次革命,不仅需耗费大量金钱,还会动用

大量人力。现有的 ERP 等企业软件当某一点业务变化时，某一点功能需要调整时，必须全部升级，这不但造成升级成本太高，而且质量无法保证。而未来 SOA 构架下的企业软件可以使某些“服务(业务组件)”不断地局部升级，新的“服务”不断地加入，只有这样的系统才能真正成为实时企业，快速适应业务变化。SOA 不仅覆盖来自于打包应用、定制应用和遗留系统中的信息，而且还覆盖来自于如安全、内容管理、搜索等 IT 架构中的功能和数据。因为基于 SOA 的应用能很容易地从这些基础服务架构中添加功能，所以基于 SOA 的应用能更快地应对市场变化，为使企业业务部门设计开发出新的功能应用。

SOA 体系结构中主要包含三个主体角色：服务请求者、服务注册者和 服务提供者，如图 3-1 所示。

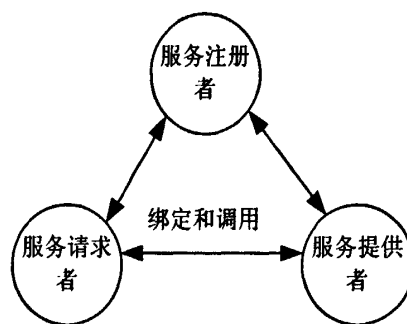


图 3-1 SOA 角色关系图

服务请求者：完成服务的查找与调用。它首先到服务注册者去查找满足特定条件的、可用到的服务，一旦找到，服务请求者将绑定到服务提供者并进行实际的服务调用。

服务注册者：集中存储服务信息，方便服务请求者查找服务。同时为服务提供者提供一个发布服务信息的平台。服务注册者实际起到了中介的作用。

服务提供者：通常情况下，服务提供者就是服务拥有者，它负责将服务信息发布到服务注册中心，同时要控制对服务的访问、执行以及服务的维护和升级。

随着 Web Service 技术的推出和应用，SOA 的思想被不断的应用到实际中，尽管 SOA 不是 Web Service，但是 Web Service 是目前最适合实现 SOA 的技术

3.1.1 Web Service

Web Service 提供给使用者某种业务或者某种数据访问的服务，并且

以管制的模式发布这些服务。它是基于 Internet, 分布式的技术。Web Service 技术体系由一组开放的、业界通用的标准和规范构成。它建立在非常成熟的技术基础上, 通过基于 XML 技术的 Web Service 描述语言 (WSDL) 对提供的服务进行封装, 然后通过统一的定义、查找和集成协议(UDDI)在 Internet 发布该服务并且提供服务的企业和组织的相关信息, 使服务使用者或者潜在的客户可以用一些标准的方式访问。它最大优势就是提供了跨平台的、由不同语言编写的应用程序之间交互的一系列标准。

Web Service 服务技术组件是一套开放的规范, 基本部分包含 HTTP、XML、SOAP、WSDL、UDDI。HTTP 协议是最底层的, 也是一个被广泛应用的、类似 RPC 的简单协议, 并且是对防火墙友好的。XML 扩展标记语言以通用的方式表示数据, 它已经被广泛使用。SOAP 协议是一个基于 XML 的消息传递协议, 它对不同平台和语言都有很好的支持。它同时支持消息传递和请求/响应通信模型。SOAP 与其他类似消息传输协议一样都需要使用接口定义语言 IDL(Interface Definition Language)。它使用的 WSDL 是一个基于 XML 的服务接口定义语言, 定义了服务接口和其实现特征。除了开放的、被广泛采用的标准外, Web Service 解决方案还能提供高度的互操作性和访问的安全性。

web 服务协议的组成体现出了一种栈式的结构。如图 3-2 栈形结构所示。

服务发布/发现	UDDI
服务描述	WSDL
XML 协议	SOAP
网络传输	HTTP、SMTP、FTP、TCP/IP

图 3-2 栈形结构

(1)传输网络层。网络传输层也是 Web Service 协议栈的最底层, 它包含了各种应用广泛的信息交换协议技术。使用者以基本的 HTTP, FTP 等协议收发信息。信息的内容可以是各种格式, 信息收发者相互告知信息中数据的含义便能完成数据交换。

(2)XML 消息层—SOAP。构建在传输网络层上的就是 XML 消息层。而 Web Service 通常使用的是基于 XML 技术的 SOAP 协议。这一层指定了应用程序通信所使用的消息格式。

(3)服务描述层—WSDL。服务描述层给服务提供者提供了一种机制即 WSDL, 可以描述他们的 Web 服务有哪些功能。WSDL 文档描述 Web 服务具有一个端点或者端口的集合, 这些端口对面向文档和面向过程的消息

进行独立的操作。WSDL 对于 Web 服务的作用就如同 COBRA-IDL 对于 CORBRA 的作用一样。WSDL 只是一种描述手段，要说明业务环境，服务质量，服务之间的关系还需要其他的描述手段协助。

(4)服务发布和发现层——UDDI。服务发布者可以用很多方式把 WSDL 文档发给服务消费者，比方说通过电子邮件，也可以将 WSDL 文档和其他相关信息发布到服务注册中心，发现和发布商业信息的规范就叫做 UDDI。服务消费者可以通过服务注册中心找到自己需要的服务以及和服务相关的更为详实的资料。

3.1.2 .NET 对 SOA 的支持

J2EE 和 .NET 是两个 SOA 支持平台，.NET 是本文选择实现系统的技术，故在此仅讨论 .NET 对 SOA 的支持。.NET 与 SOA 相关的一个主要部分是 ASP.NET 环境，它在 SOA 中的交付 Web 技术层之内（且进一步由 Web 服务增强（Web Services Enhancements, WSE）扩展所补充）。

ASP.NET, WSE 及程序集和 .NET 框架的其他部分相互关联提供了一个平台，通过该平台能够创建支持 SOA 的服务。如图 3-3 所示。

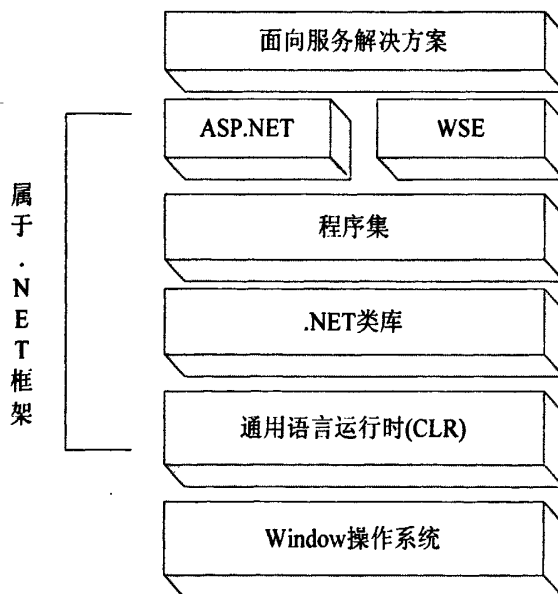


图 3-3 与 SOA 关联的 .NET 框架相关层

下面说明一个 .NET Web 应用程序在服务提供者，服务请求者，服务代理模型中如何定位。

(1)服务提供者

.NET 服务提供者是作为一个特殊 ASP.NET 应用变种而存在的 Web 服

务,称为 ASP.NET Web 服务。可以使用识别担当端点服务部分的“.asmx”扩展,来辨别一个指向 ASP.NET Web 服务的 URL。ASP.NET Web 服务能够独立作为一个包含内联代码及特定的 ASMX 文件存在,但它们通常包含一个 ASMX 端点及一个驻留各个业务逻辑的编译程序集。

在.NET 服务提供者中的业务逻辑通常由一个编译程序集表示。

ASMX 文件引用程序集并包含所有信息,以处理提供者 WSDL 并对去向和来自业务逻辑的请求与响应进行转换。

ASP.NET Web 服务的执行环境由 ASP.NET 和相关类库(API)所建立。

(2)服务请求者

为支持服务请求者的创建,.NET 提供了一个位于服务请求者的应用逻辑端代理类,并复制服务提供者接口,这允许服务请求者与本地代理类交互,同时将所有远程处理与消息编组活动那个委托给逻辑。.NET 代理将方法调用转换成 HTTP 请求,随后将由服务提供者处理的消息转换成本地方法回调。

(3)服务代理

ASP.NET 环境利用许多执行各种运行时处理任务的系统级代理。ASP.NET 运行时要配备一系列带有 HTTP 模块的 HTTP 管道。这些服务代理能够执行系统任务,比如认证、授权和状态管理。也能够创建定制 HTTP 模块,在端点联系之前和之后执行各种处理任务。

3.2 系统总体架构

本文第二章对飞信机器人系统的详细需求分析确定了飞信机器人系统的实现目标。系统的总体架构分为物理架构,逻辑架构,系统的运行机制三个角度来进行设计。

从飞信网络的角度来看,飞信机器人系统是飞信机器人的集合,它处于飞信服务器的客户端的位置,本节对飞信机器人系统的物理架构进行了详细的描述。

在系统逻辑结构设计上,飞信机器人系统一方面需要拥有在不影响系统架构下不断增加新服务的能力,另一方面需要能够在多用户并发访问的情况下,系统保持良好的稳定性。本文采用分层的思想将开发的重点分散到各个层面去完成,各层之间为松散耦合的关系,系统的核心层(系统服务层)集中了所有的业务,采用面向服务(SOA)的体系结构^[25-35]来设计,使业务的更新不需要改变系统的体系结构。

下面分别对系统的物理架构,逻辑架构,系统的运行机制作介绍。

3.1.1 系统物理架构

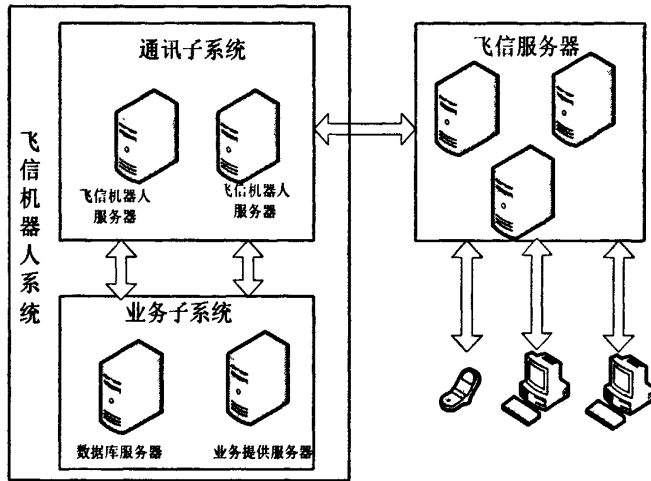


图 3-4 飞信机器人系统物理架构

上图描述了飞信机器人系统的物理架构，从飞信网络的角度来看，飞信机器人系统是飞信机器人的集合，所有的飞信机器人都是飞信服务器的客户端，都是用户联系人列表上的一个联系人，通过飞信服务器和用户通讯，处于客户端的地位。从互联网的角度来看，每一个飞信机器人服务器都为数万个客户服务，而整个飞信机器人系统是飞信机器人服务器的集群，系统处于服务器的地位。所以飞信机器人系统是处于客户端地位的服务器。

在飞信机器人系统内部分为通讯子系统和业务子系统。在实际运行中，通讯子系统会将用户的请求转给业务子系统，业务子系统会将回复发给通讯子系统，它们之间需要进行频繁的通信，且两个子系统之间的通信延迟不能够影响系统的性能，这对两个进程之间的通信机制提出了较高的要求，选用机制的好坏直接影响到系统的性能，是本课题设计的一个关键。为了保证系统的性能，本课题打算将通讯子系统所在服务器和业务子系统所在服务器部署在同一局域网内，.NET 框架提供了几种好的选择来完成进程间通信（IPC）：Web Service, Remoting, 管道。Web Service 的特点是平台独立性、跨语言（只要支持 XML 语言都可以）以及能穿透企业防火墙 Remoting 很快，但是它需要部署一台 Web Server，而且速度比较慢，不能满足通讯子系统和业务子系统之间传输速度要求，Remoting 与 Web Service 有所不同的是，它支持 HTTP 以及 TCP 信道，而且它不仅传输 XML 格式的 SOAP 包，也可以传输传统意义上的二进制流，这使得它变得更高效也更加灵活。但是 Remoting 的序列化增加了处理过程，使得部署在本地的

两个进程通信增加了不必要的处理步骤, 在本课题部署的特定环境中, 还是慢了些。 .NET 提供的命名管道是一个比较好的选择, 不会进行二进制序列化, 所以提供了更快的 IPC。这个解决方案最有效的使用是在一个应用程序需要和另一个应用程序进行非常频繁的、短文本的消息通信的情况下, 并且是在同一台机器或在同一局域网内部。对于结构化的数据交换, 这些文本消息也可以是 XML 文档或序列化的 .NET 对象。通信没有安全层, 因为命名管道最多智能在局域网中运行, 所以假定安全问题由别的层处理。

在本课题的实现中, 通讯子系统和业务子系统之间采用命名管道通信, 业务子系统直接和通讯子系统直接通信的部分要和通讯子系统部署在同一个局域网内, 但是对数据库服务器和具体的业务提供服务器的部署位置没有要求。

3.1.2 系统逻辑结构

软件设计中的模块化设计和面向对象的设计方法可以取得局部优化的效果, 但是目前的软件系统随着功能越来越多, 整个系统也越来越庞大, 局部优化的技术难以实现整个系统的优化(包括代码的优化, 对设计时间、费用以及可靠性、兼容性和可维护性等方面的综合考虑)。需要有更高层次的软件设计框架来完成系统设计。在软件体系架构设计中, 分层结构目前是最常见, 也是最重要的一种结构。整个软件系统根据其所需完成的功能和任务, 分为相对独立的若干层, 每一层均可以是独立的进程、线程或其它程序模块。一个好的分层式结构, 可以使得开发人员的分工更加明确。一旦定义好各层次之间的接口, 负责不同逻辑设计的开发人员就可以分散关注, 齐头并进。例如 UI 人员只需考虑用户界面的体验与操作, 领域的设计人员可以仅关注业务逻辑的设计, 而数据库设计人员也不必为繁琐的用户交互而头疼了。每个开发人员的任务得到了确认, 开发进度就可以迅速的提高。

松散耦合的好处是显而易见的。如果一个系统没有分层, 那么各自的逻辑都紧紧纠缠在一起, 彼此间相互依赖, 谁都是不可替代的。一旦发生改变, 则牵一发而动全身, 对项目的影响极为严重。降低层与层间的依赖性, 既可以良好地保证未来的可扩展性, 在复用性上也有明显的优势。每个功能模块一旦定义好统一的接口, 就可以被各个模块所调用, 而不用为相同的功能进行重复地开发。进行好的分层式结构设计, 标准也是必不可少的。只有在一定程度的标准化基础上, 这个系统才是可扩展的, 可替换

的。而层与层之间的通信也必然保证了接口的标准化。综上所述，分层结构的主要优点是：

- 1、开发人员可以只关注整个结构中的其中某一层；
- 2、可以很容易的用新的实现来替换原有层次的实现；
- 3、可以降低层与层之间的依赖；
- 4、有利于标准化；
- 5、利于各层逻辑的复用。

按照分层的思想，将飞信机器人系统的逻辑结构分为三层：应用层、系统服务层、数据存储层。三层之间是低耦合的关系，可以独立开发。系统的逻辑结构如图 3-5 所示。

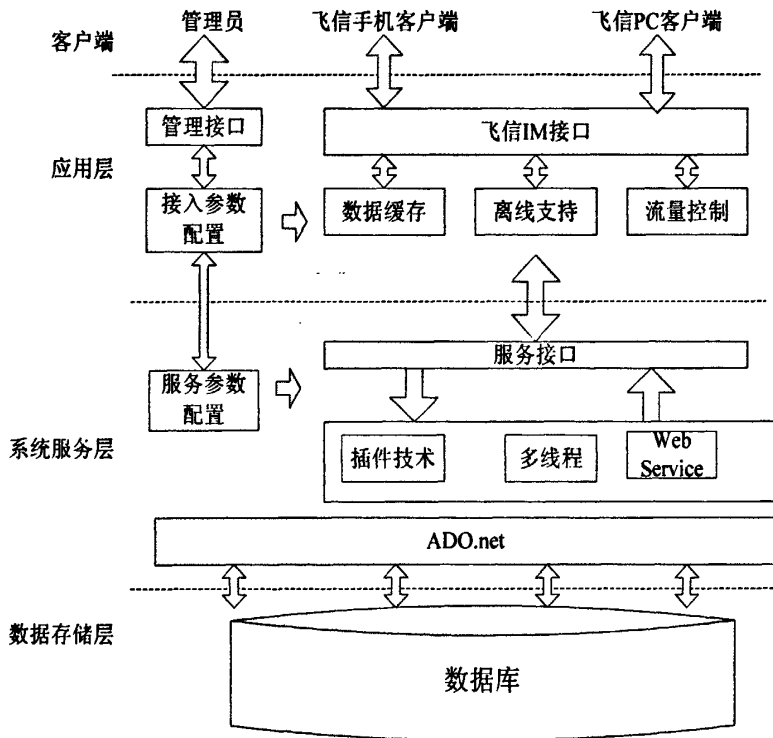


图 3-5 系统逻辑结构图

应用层：和飞信服务器进行信息交互，实现飞信 IM 通讯接口，并且具备数据缓存，离线支持，流量控制的功能。这一层中集中解决飞信客户端大部分功能的实现问题，所有与之相关的影响底层通讯性能的部分均在这一层解决。

系统服务层：核心控制层，这一层采用 SOA 体系结构进行设计，在实现中，借助.NET 的相关技术来达到实现面向服务框架的架构的目的，在本课题中，采用多线程技术和插件技术，调用.net 框架支持的 Web Service 方法，来实现业务处理和通讯的接口。通过.NET 提供的 ADO.NET 来实现

和数据存储层进行交互。

数据存储层：用于飞信用户的个人信息，状态信息，日志等信息数据的存储。在这一层主要考虑数据库的设计。

其中系统服务层是整个系统的核心，采用 SOA 体系结构设计，在物理架构中系统服务层和数据存储层对应业务子系统，在 3.4 节中将作进一步的描述。

3.1.3 系统的运行机制

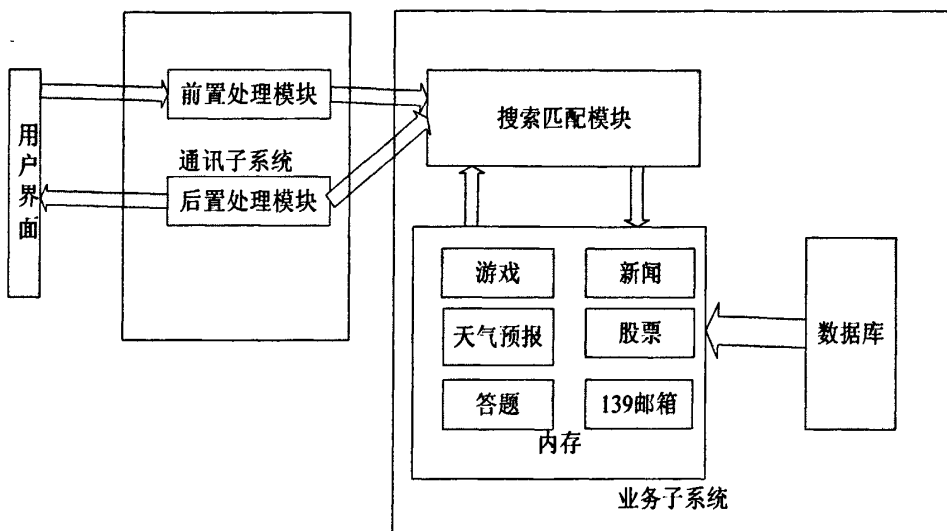


图 3-6 系统的运行机制

系统功能模块如上图 3-6 所示。

通讯子系统负责飞信机器人服务连接与断开，以及服务连接建立后用户与机器人之间的会话管理。分为如下三种情况：

(1) 用户打开与飞信机器人的对话窗口时，前置处理模块自动检测到这一动作，触发搜索匹配模块搜索事件发生，返回主功能菜单给后置处理模块，触发后置处理模块下行向用户推送飞信机器人的功能主菜单。

(2) 用户上行向飞信机器人发送消息，触发前置处理模块将该消息上行传递到搜索匹配模块，根据搜索算法，或者直接返回相应消息，或者转向具体的插件进行处理，之后下行传递数据给后置处理模块，触发后置处理模块下行向用户推送数据。

(3) 前两种情况均属于由用户先发起的被动式返回消息，最后一种情况是业务子系统主动下发短信给后置处理模块，触发后置处理模块下行发消息给指定用户。短信的内容，下发时间，下发的对象均可由管理员自

行定义。

业务子系统由搜索匹配模块，具体的功能子集和数据库组成。

搜索匹配模块主要负责将用户发送的指令定位到具体的功能子集，目前该算法的基础是菜单式搜索，当定位到具体的功能子集，比如游戏，则交由游戏子集作下一步处理，之后将结果返回给搜索匹配模块，由其进行下一步处理，为了能够能使系统的响应速度更快，有更好的扩展性，加强搜索的算法也是下一步的研究方向。

功能子集由一系列不同的功能插件组成，它们有易于开发的二次接口，使程序员可以只关注具体业务的开发而不用考虑下层的平台到底是什么。这些功能插件可以直接和 Web service 相结合。实现服务的可扩展性。数据库存放了与信息相关的所有数据，它影响着整个飞信机器人系统的性能，随着业务的增多，信息存取快速的增长，数据库性能的优化在系统达到一定用户量后，可以预见也是重要的研究方向。

3.3 通讯子系统架构

通讯子系统负责系统的通讯，关键点在于功能模块的划分，协议的封装，及网络通讯。本小节将从通讯子系统功能结构设计，通讯子系统处理时序图两个方面来描述通讯子系统架构。

3.3.1 通讯子系统功能结构

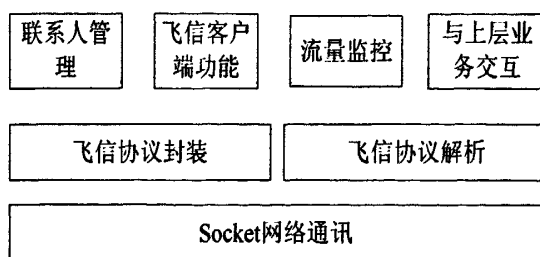


图 3-7 通讯子系统功能结构

通讯子系统主要完成通讯功能，从图 3-7 中可看出，通信子系统按照网络模型划分功能结构分为三个层次。Socket 网络通讯是最底层，主要实现 TCP 的承载，实现双向可靠传输，飞信协议在 TCP 之上，飞信协议封装，飞信协议解析模块按照飞信协议的细节分别进行封装和解封，具体的功能建立在下两层之上，来完成联系人管理，飞信客户端功能，流量监控以及与上层业务的交互。通信子系统划分的具体功能模块如下图 3-8 所示。

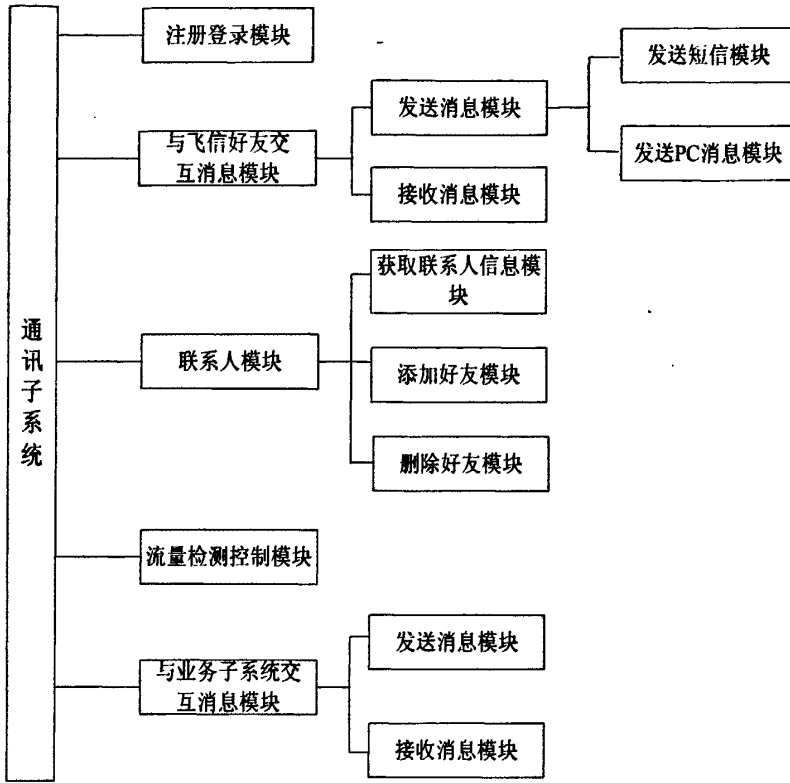


图 3-8 通讯子系统功能模块图

3.3.2 通讯子系统处理时序图

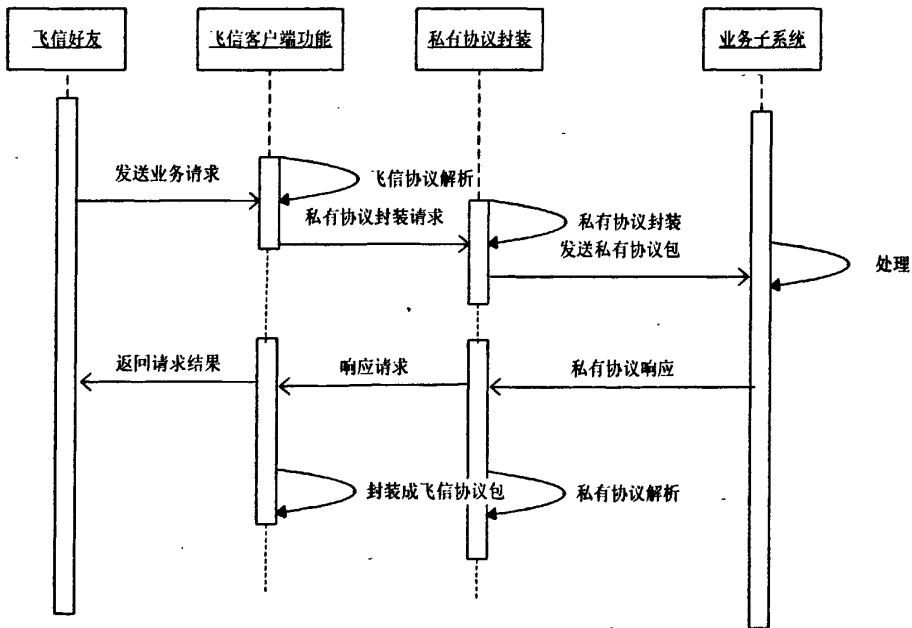


图 3-9 通讯子系统处理时序图

3.4 业务子系统架构

本小节将从业务子系统逻辑结构设计，功能结构设计，通讯子系统处理时序图三个方面来描述业务子系统架构。

3.4.1 业务子系统逻辑结构

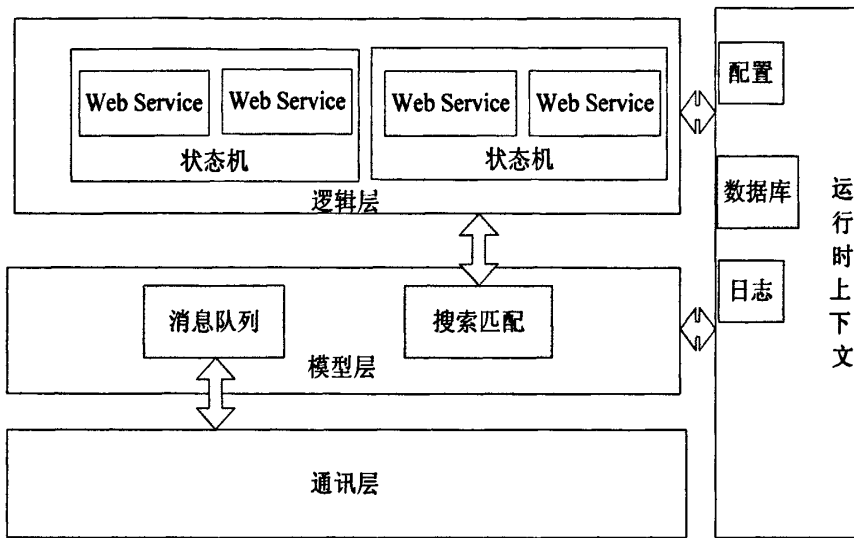


图 3-10 业务子系统逻辑结构图

如图 3-7 所示，业务子系统由三层架构组成，由下到上分别是通讯层，模型层，逻辑层。

通讯层负责与通讯子系统的通讯，接收通讯子系统发来的消息，并将消息通过消息队列转发给模型层，并且通过消息队列接收模型层回复的消息，再按照与通讯子系统约定的格式转发给通讯子系统。

模型层是一个中间层，关联通讯层和逻辑层，从通讯层传上来的数据，按照消息类型不同（短信，PC 消息，发送者当前状态），为每个消息类型建立消息队列，消息队列可以协调网络速度和业务处理速度。从通讯层传上来的数据进入搜索匹配过程，根据业务不同，进入逻辑层的不同状态机进行处理。

逻辑层封装了大量的业务，内部是按照业务划分有限状态机，将有限状态机设计成为具有相同接口的插件，在插件中处理具体的业务，如果添加新业务，可以重写插件。通用插件中引入 Web Service,和服务提供商共同协作，可以作为服务请求者来调用服务提供者提供的 Web 服务，或者作为服务提供者来提供服务，实现互操作性，极大提高系统可扩展性。

运行时上下文提供配置，数据库，日志等一系列公共服务。

3.4.2 业务子系统功能结构

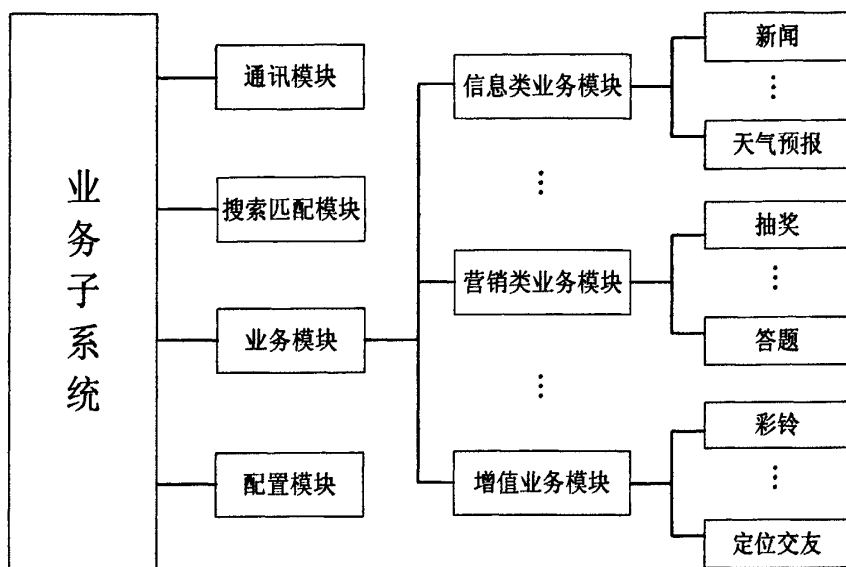


图 3-11 业务子系统功能模块图

从功能上业务子系统划分为通讯模块，搜索匹配模块，业务模块和配置模块。

通讯模块的功能是完成和通讯子系统的信息交互，即将通讯子系统发送的消息进行解析，分类成 PC 消息，短信消息，同时将用户状态（手机客户端，PC 客户端）一并发送给搜索匹配模块，并且将搜索匹配模块返回的消息按照封装成和通讯子系统约定好的格式，发送给通讯子系统。

搜索匹配模块的功能是识别飞信用户的需求，将需求定位到相应的业务模块处理，搜索匹配模块是与飞信用户进行自然语言交流的关键，如何从用户的消息中提取出关键词，并且定位到相应的业务模块是其所要解决的问题，在本课题中采用的是菜单的方式将系统所支持的业务组织起来，系统可以识别输入菜单编号，业务名称简写，从而定位到合适的业务模块。

业务模块实现具体业务需求，将所能提供的业务按照大的种类分为信息类业务模块（新闻，天气预报等），营销类业务模块（抽奖，答题等），增值业务模块（彩铃，定位交友等）。所有的互联网应用都可以作为一个小的业务模块在不影响系统的整体架构条件下加载到系统中。

配置模块的功能是配置用户与飞信机器人对话的界面，包括功能菜单，皮肤等。配置模块实现插件的动态加载，并且通过自定义配置文件来决定显示给飞信用户的菜单的样式与格式。

业务子系统的功能满足 2.1.3 节业务子系统的需求。

3.4.3 业务子系统处理时序图

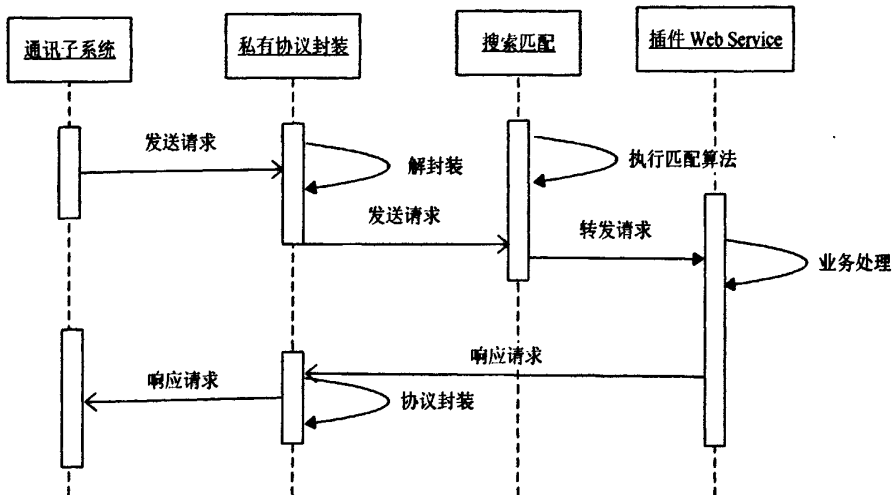


图 3-12 业务子系统处理时序图

3.5 本章小结

本章简要介绍了先进的 SOA 技术，在第二章需求分析的基础上，将 SOA 架构引入飞信机器人系统的业务子系统，阐述了飞信机器人系统的总体架构，通讯子系统架构以及业务子系统的架构，并对其中用到的关键技术进行了描述。

第四章 系统实现与应用

在飞信机器人系统的两个子系统通讯子系统和业务子系统实现过程中,有一些关键问题需要解决,本章对这些问题的解决方案做了阐述。另外,本章还对当前飞信机器人系统的应用情况做了简要的介绍。

4.1 通讯子系统的实现

尽管目前即时通讯已经有了诸如即时信息和空间协议(IMPP)、针对即时通讯和空间平衡扩充的进程开始协议 SIP(SIMPLE)以及 XMPP 等标准协议^[36-38],但是各主流即时通讯厂商为了争夺用户,并没有采用标准协议,而是定义了自己的私有协议。飞信协议是中国移动自己定义的私有即时通讯协议,系统基于一些得到广泛支持的协议进行构建,并对一些协议进行了简化和扩展,以便适应移动通讯技术的网络特点,并支持特定的服务。基础协议由 SMS, HTTP, XML, 加密算法, SDP, SIP-C 组成, SIP-C 为飞信通信的基础协议,遵循中国移动自定义的 SIP-C 协议规范。本节通过搭建协议分析环境,分析了飞信协议的通讯过程,剖析了大部分飞信客户端功能。本节将简要介绍 SIP-C 协议规范,分析飞信的基本传输过程,并介绍本课题对飞信客户端功能的实现方法。

4.1.1 SIP-C 简要介绍

SIP-C (Compact SIP) 是一个简化和增强的 SIP 协议,以适应移动终端设备的接入特点,但会话的逻辑过程与 SIP (RFC3261^[39]规定)及相关协议确立的逻辑过程保持不变。

4.1.1.1 会话建立的过程

SIP 是一个基于事务 (Transaction) 处理的协议,部件之间的交互是通过一系列无关的消息交换所完成的。事务是指一个发送方和接收方的交互过程,由请求和应答组成。请求分为 INVITE 类型和非 INVITE 类型。应答分为响应型的应答和确认型的应答。响应型的应答是指这个应答仅代表对方收到请求。请求经过处理后都必须返回确认型的应答。响应型的应答有 1XX, 确认型的应答包括 2XX、3XX、4XX、5XX 和 6XX。一个事务由一个请求一个或多个响应型应答、一个确认型应答组成。

下面描述 INVITE 和非 INVITE 情况下的典型消息流程：

INVITE 消息流程如图 4-1 所示：

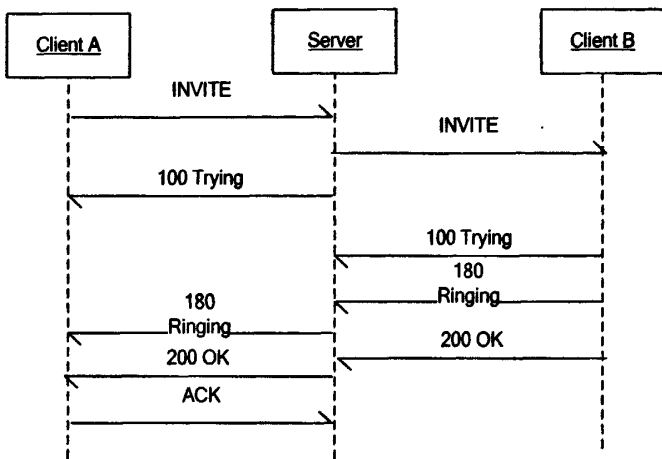


图 4-1 INVITE 消息流程图

非 INVITE 消息流程如图 4-2 所示

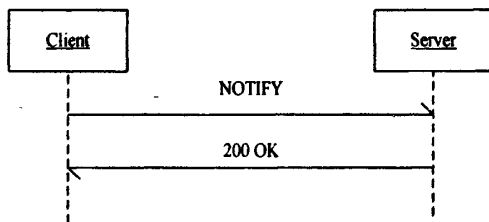


图 4-2 非 INVITE 消息流程图

4.1.1.2 SIP-C 方法缩写

SIP-C 定义了对方法的缩写表示方式，SIP-C 消息必须使用定义的缩写。图 4-3 列出了 SIP-C 方法对 SIP 方法的简化，具体方法的用途在此不再描述，可参考 RFC3261。

表 4-1 SIP-C 方法缩写

SIP 方法	SIP-C 方法
ACK	A
BYE	B
BENOTIFY	BN

CANCEL	C
INVITE	I
MESSAGE	M
NOTIFY	N
NEGOTIATE	NEG
REGISTER	R
REFER	REF
SERVICE	S
SUBSCRIBE	SUB
OPTIONS	O
INFO	IN

4.1.1.3 SIP-C 消息头缩写

SIP-C 定义了对消息头的缩写方式, SIP-C 消息中的消息头必须使用缩写方式。图 4-4 列出了 SIP-C 消息头对 SIP 消息头的简化, 具体消息头的用法在此不再描述, 可参考 RFC3261。

表 4-2 SIP-C 消息头缩写

SIP 消息头	SIP 简写方式	SIP-C 简写方式
Authorization		A
Content-Type	C	
Content-Encoding	E	
CSeq		Q
Call-ID	I	
Content-Length	L	
Contact	M	
Date		D
Event		N
Expires		X
From	F	
Subject	S	
Supported	K	
To	T	
Via	V	

WWW-Authenticate		W
Refer-To		RT
Referred-By		RB
Require		RQ
Unsupported		UK
Source		SO
X-MESSAGE-ID		XI

4.1.2 飞信的基本传输协议及通信过程

飞信通信的系统结构如图 4-3 所示

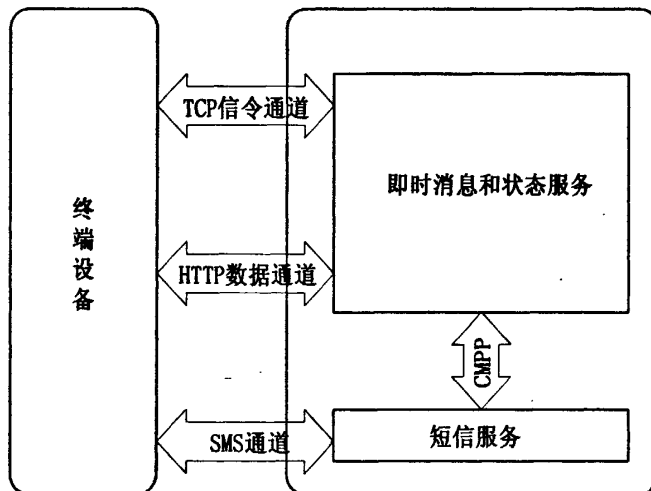


图 4-3 飞信通信的系统结构

在飞信的通信中，使用 TCP 作为传输层承载协议，用 HTTP 来获取系统的配置，验证用户的身份等，在用户的登录过程中，使用了 MD5 加密算法。在用户登录到飞信服务器后，使用 SIP-C 作为飞信通信的承载协议，SIP-C 使用 TCP 作为传输层承载协议。终端使用 SDP 协议来描述会话类型，建立各种类型的会话。在通信中，XML 作为主要的数据交换格式。下面将对飞信的登录，通信，添加/删除联系人，及注销过程作详细的分析。

4.1.2.1 注册登录

飞信账号注册到飞信服务器需要经过如下几次交互过程：

第一步，发送 HTTP 消息获取系统配置信息，格式如下：

POST /nav/getsystemconfig.aspx HTTP/1.1

User-Agent: IIC2.0/PC 2.1.0.0

Host: nav.fetion.com.cn

Content-Length: 233

Connection: Keep-Alive

得到返回的配置信息，从中提取出 sipcSslProxy，ssiAppSignIn，ssiAppSignOut，portal，serverVersion 对应相关配置信息。

第二步，用 get 方法发送如下消息：

https://uid.fetion.com.cn/ssiportal/SSIAppSignIn.aspx?mobileno=<手机号>&domains=fetion.com.cn%3bm161.com.cn&digest=<手机号, 密码 MD5 加密>

从服务器返回的 cookie 中得到 ssid 值。从返回的结果里找出如下匹配项：

```
<user uri="sip:xxxxxxxxx@fetion.com.cn;p=xxxx" mobile-no="xxxxxxxxxxx"
user-status="101" user-id="xxxxxxxxx">
```

将@fetion.com.cn 前的飞信号提取出,记为 Fetion number。

第三步，发送如下格式数据包给 sipc 代理服务器。

R fetion.com.cn SIP-C/2.0

F: <Fetion number>

I: 1yy

Q: 1 R

L: 281

```
<args><device type="PC" version="0" client-version="3.3.0370" /><caps
value="simple-im;im-session;temp-group;personal-group" /><events
value="contact;permission;system-message;personal-group" /><user-info
attributes="all" /><presence><basic value="400" desc=""
/></presence></args>
```

服务器返回如下格式数据包：

SIP-C/2.0 401 Unauthorized

F:xxxxxxxxx

I:1

Q:1R

W:Digest algorithm="MD5-sess;SHA1-sess", nonce="xxxxxxxxxxxxxxxxx"

从返回的数据报里提取 nonce 的值。

第四步，计算 cnonce 值，cnonce 的计算方法是用了 4 个随机数组成一个 16 进制的字符串。

计算 response 值，具体步骤如下：

a. 求 "sid:domain:password" 这个 utf8 字符串的哈希值 ("sid:domain:password"的 MD5 值), 得出 key。

b.求"key:nonce:cnonce"的 MD5 值, 得出 h1。

c.求"REGISTER:sid"的 MD5 值, 得出 h2。

d.求"h1:nonce:h2"的 MD5 值, 即为 response。

第五步, 发送请求认证, 将 2, 3, 4 步得到的 response,cnonce,ssic 下面消息格式中:

R fetion.com.cn SIP-C/2.0

F: xxxxxxxxxx

I: 1

Q: 2 R

A:Digest

algorithm="SHA1-sess",response="<response>",cnonce="<cnonce>",salt="072D6401",ssic="<ssic> "

L: 281

<args><device type="PC" version="0" client-version="3.3.0370" /><caps

value="simple-im;im-session;temp-group;personal-group"

/><events value="contact;permission;system-message;personal-group"

/><user-info attributes="all" /><presence><basic value="400" desc=""

/></presence></args>

第六步, 认证通过后登录成功。

4.1.2.2 注销

当飞信机器人退出时, 需要发送注销请求消息, 收到返回的注销成功应答消息后退出。

发送注销请求消息数据报格式如下:

R <Domain> SIP-C/2.0

F: <SID>

X: 0

I:< call-id-value>

Q: <cseq-value>R

发送带有值为 0 的 Expires 头的 REGISTER 消息, 来从系统中注销登陆。

注销成功, 正常的注销应答消息格式如下:

SIP-C/2.0 200 Ok

X: 0

I: <call-id-value>

Q: <cseq-value> R

客户端需等待服务器应答消息返回后再退出，如果客户端没有收到正常的注销应答消息，至少要等待 30 秒，如果超时时仍无应答，则可以自动退出。即使返回的状态码不为 2XX，可忽略错误，正常退出。

4.1.2.3 发送、接收消息

飞信建立会话的过程符合 SIP 建立会话的标准过程。

4.1.2.4 添加联系人

添加联系人需向服务器发送请求消息，数据包格式如下：

S <Domain> SIP-C/2.0

F: <SID>

N: Add Contact

I: <call-id-value>

Q: <cseq-value> S

L: <content-length-value>

```
<Root xmlns="http://tempuri.org/DateExchange.xsd">
```

```
  <deltaNum><delta-number-value></deltaNum>
```

```
  <uri><contact-URI></uri>
```

```
</Root>
```

delta-number-value 为当前联系人信息的版本号。操作必须给出正确的版本号，否则操作将直接失败。

contact-URI 为要添加的联系人的 URI，格式如下：

sip:SID@Domain

服务器在正确情况下返回应答消息的数据包格式如下：

SIP-C/2.0 200 OK

I: <call-id-value>

Q: <cseq-value> S

因为每个联系人列表和组的操作都需要当前的联系人信息版本号 (**delta-number-value**)，所以不能同时进行多个相关这些数据的操作。如果前一个操作返回 200 OK，下一个操作需要等待相应的 NOTIFY 消息回来，以获得下一个有效的版本号，如果使用了错误的联系人信息版本号，服务器会返回 409 Conflict 应答，说明此时客户端的联系人信息与服务器端不一致，此时客户端，重新订阅联系人信息，先取消上一次订阅，再使用一个新的会话(新的 Call-ID)重新订阅。

4.1.2.5 删除联系人

删除联系人需向服务器发送请求消息，数据包格式如下：

```
S <Domain> SIP-C/2.0
F: <SID>
N: DeleteContact
I: <call-id-value>
Q: <cseq-value> S
L: <content-length-value>
<Root xmlns="http://tempuri.org/DateExchange.xsd">
  <deltaNum><delta-number-value></deltaNum>
  <uri><contact-URI></uri>
</Root>
```

delta-number-value 为当前联系人信息的版本号。操作必须给出正确的版本号，否则操作将直接失败。

contact-URI 为要删除的联系人的 URI，格式如下：

```
sip:SID@Domain
```

收到的正常应答消息数据包格式如下：

```
SIP-C/2.0 200 OK
I: <call-id-value>
Q: <cseq-value> S
```

因为每个联系人列表和组的操作都需要当前的联系人信息版本号 (**delta-number-value**)，所以不能同时进行多个相关这些数据的操作。如果前一个操作返回 200 OK，下一个操作需要等待相应的 NOTIFY 消息回来，以获得下一个有效的版本号，如果使用了错误的联系人信息版本号，服务器会返回 409

Conflict 应答, 说明此时客户端的联系人信息与服务器端不一致, 此时客户端, 重新订阅联系人信息, 先取消上一次订阅, 再使用一个新的会话(新的 Call-ID) 重新订阅。

4.1.3 飞信客户端功能的实现

4.1.3.1 基本协议通讯的实现

飞信客户端与服务器通信实现的是TCP, HTTP承载。用HTTP来获取系统的配置信息, SIP-C协议(TCP承载)作为飞信的通信协议。下面对这两种协议通讯的实现作介绍。

(1) HTTP通讯

HTTP (Hyper Text Transfer Protocol) 是现代互联网上最重要的一种网络协议, 位于TCP/IP协议的应用层, 是一个面向无连接、简单、快速的C/S结构的协议。HTTP的工作过程大体上分连接、请求、响应和断开连接四个步骤。C#语言对HTTP协议提供了良好的支持。

HttpWebRequest类实现了很多通过HTTP访问WEB服务器上文件的高级功能。将发送到Internet资源的公共HTTP标头的值公开为属性, 由方法或系统设置, 常用的由属性或方法设置的HTTP标头为: 接受, 由Accept属性设置, 连接, 由Connection属性和KeepAlive属性设置, Content-Length, 由ContentLength属性设置, Content-Type, 由ContentType属性设置, 范围, 由AddRange方法设置。实际使用中是将标头信息正确设置后, 传递到WEB服务器, WEB服务器根据要求作出回应。

HttpWebResponse类专门处理从WEB服务器返回的HTTP响应, 这个类实现了很多方法, 具有很多属性, 可以全面处理接收到的互联网信息。在HttpWebResponse类中, 对于大多数通用的HTTP标头字段, 都有独立的属性与其对应, 程序员可以通过这些属性方便的访问位于HTTP接收报文标头字段中的信息。

本课题对HttpWebResponse, HttpWebRequest进行了进一步封装, 将所有Http的连接封装在HttpHelper类中。

接口函数: `public static string Post(string url, string data)`

该函数完成一次post的过程, 返回值为服务器返回消息。

接口函数: `public static string[] Get(string url, string data)`

该函数完成一次get的过程, 返回值为服务器返回消息。

(2) SIP-C通讯

当登录完成后, 所有飞信用户和飞信机器人之间的通讯都使用的是SIP-C协

议, TCP承载, 实现SOCKET的网络通讯是基础, 剩下的就是按照4.1.2小节协议包数据格式进行封装和解封装的过程。

SOCKET的底层通讯实现流程见图4-4。

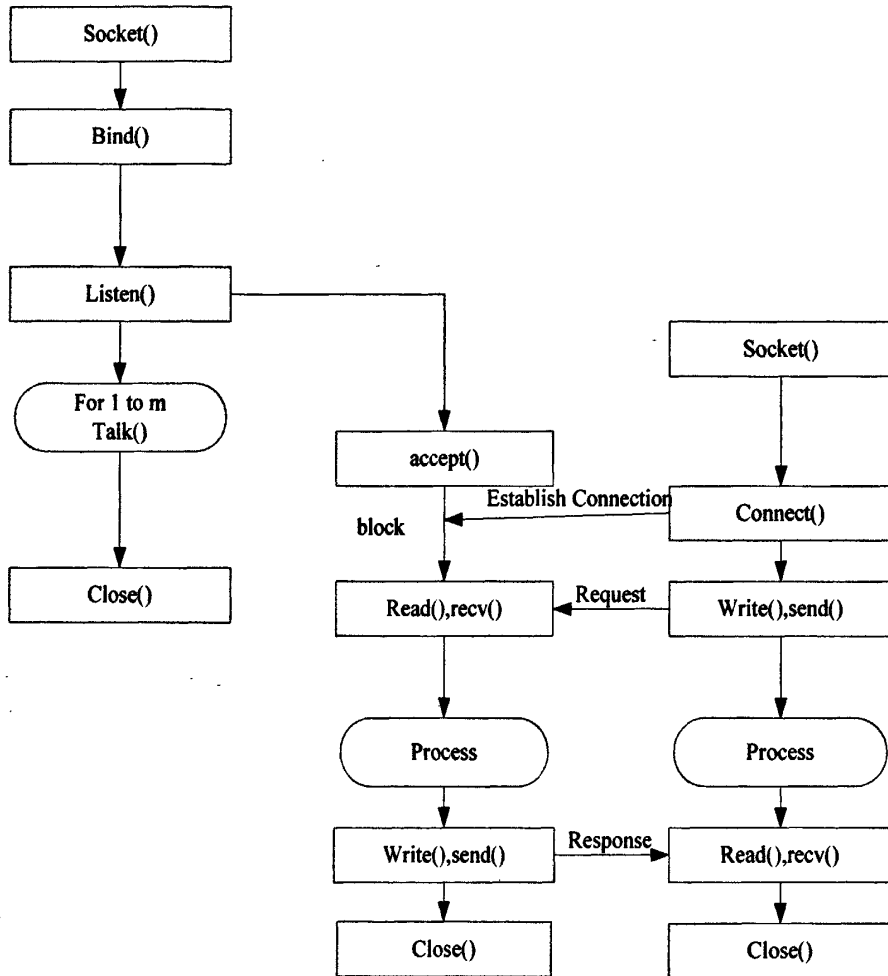


图 4-4 Socket 通讯流程图

在实现中, 将 socket 通讯的过程封装, 在 socket 层之上建立 sip-c 连接。创建 ISipconnection 抽象接口将 socket 连接建立过程进行封装, ISipconnection 的原型如下:

```

public interface ISipConnection
{
    Socket SipSocket { get; set; }
    void Connect (string ip, string port);
    int NextCallID( );
    System.Net.EndPoint LocalEndPoint { get;}
}
  
```

4.1.3.2 异常处理机制

良好的异常处理机制是衡量系统性能好坏的一个重要标准,编写软件是一项复杂的工作,即使是最好的软件也经常伴随各种各样的问题,在网络通讯中,返回的结果都不能保证百分之百成功,可能返回不成功的响应或出各种各样的差错。通常有如下两种异常:

(1) bug: 这是由程序员一方引起的错误。例如,如果进行非托管 C++编程,如果删除已分配的内存失败,从而产生内存泄漏,就会产生一个 bug。

(2) 异常: 异常往往是运行时的非正常情况,在编程时很难被估计到。异常可能包括试图连接一个已经不存在的数据库、打开已被破坏的文件、连接当前处于离线状态的机器等。任何人都无法完全控制这些异常情况。

.NET 结构化处理异常是一项很适合处理运行时异常的技术。它使开发人员现在有了统一的,而且对 .NET 领域内各种语言都相同的途径来处理错误。.NET 异常处理技术使我们不再是通过接收意义模糊的数字常量来确定问题,而是可以通过异常,它们包含容易读懂的问题描述信息和首次触发异常时调用栈的详细快照。

C#编程语言提供了 4 个允许我们引发和处理异常的关键词: try、catch、throw 和 finally。用来表示问题的类型是一个继承自(或派生自) System. Exception 的类。

本课题对所有可能出现异常的地方都设置了异常处理,来保证系统的稳定运行。在网络通信的部分,是主要引发异常的地方,主要考虑在响应不成功时候的处理机制。一种由 try {} catch {} 捕获到该异常,一种由服务器返回错误消息代码捕获到该异常。例如,如果用户登录不成功引发异常,由系统捕获到,会触发重新登录的事件。在收发信息,注销,添加联系人,删除联系人等不成功时,都会转到相应的自定义处理机制中去处理,这么做使系统的容错性大大提高。如下是登录失败引发的重新登录的事件的实现方法。

(1) 重新定义登录事件类,继承自 EventArgs 类,原型如下:

```
public class LoginEventArgs : EventArgs
{
    private string description = string.Empty;
    public LoginEventArgs()
    {
    }
    public LoginEventArgs(string description)
    {
```

```

        this.description = description;
    }
    public override string ToString()
    {
        return this.description;
    }
}

```

(2) 初始化登录事件类的对象。定义如下：

```

public event EventHandler<LoginEventArgs> LoginFailed;
EventHandler<LoginEventArgs> handlerFailed = this.LoginFailed;

```

(3) fetion 是飞信客户端 Client 类的对象。飞信整个通信过程封装在该类中，在该对象登录过程函数 public void Login(string mobile, string pwd)中定义如下错误处理机制：

```

fetion.LoginFailed += new EventHandler(fetion_LoginFailed);
fetion_LoginFailed 函数原型如下：
private void fetion_LoginFailed(object sender, EventArgs e)
{
    System.sleep(10000);
    ReLogin();//重新登录
}

```

(4) 当系统捕获到登录失败引起的异常时，将会引发相关事件。捕获的方法如下：

```

catch (Exception ex)
{
    if (handlerFailed != null)
    {
        handlerFailed(this, new LoginEventArgs(ex.ToString()));
    }
}

```

上述是登录失败会在10s后重新登录的完整描述，对收发信息，注销，添加联系人，删除联系人不成功采用类似的处理机制。

4.2 业务子系统的实现

业务子系统主要任务是通过和通讯子系统的消息交互，来实现和飞信

用户的智能对话，业务子系统需要重点解决两方面的问题，第一，服务提供者如何能快速的将服务集成到飞信机器人系统上，即飞信机器人系统必须实现简单的二次开发接口，使开发人员在具体的业务实现中，不影响系统的整体架构。第二，如何实现和飞信用户的智能对话，来让飞信用户感觉是在和一个和善，服务态度好的真实的客服人员或者是网友在对话。本节对系统的业务流程作了介绍，然后对解决上述两个问题的关键技术作了介绍。

4.2.1 业务流程

业务子系统总的业务流程如图 4-5 所示。

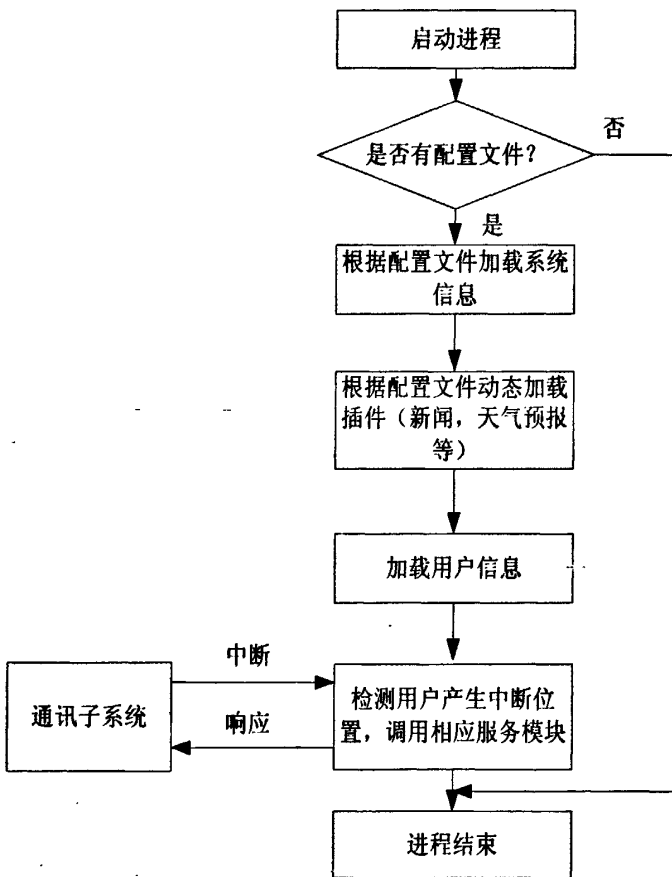


图 4-5 业务子系统业务流程图

当业务子系统进程启动后，首先将读取配置文件，如果配置文件出错或者不存在，进程直接结束。如果配置文件正确，则按照配置文件初始化系统信息，动态加载插件，最后从用户数据库中加载用户信息到缓存中，全部启动完毕后，系统进入休眠状态，通讯子系统向业务子系统的请求消息触发了业务子系统进

入搜索匹配模块进行业务的查找,定位到相应的业务插口时,进行具体的业务处理,每一次通讯子系统与业务子系统的通迅过程创建一个线程,当处理完成后,该线程被释放,飞信用户当前的状态,发生事件的记录均记录在数据库中,当飞信用户再次与飞信机器人系统通讯时,业务子系统将飞信用户当前的状态读出,接着进行上次的处理。当系统关闭时,会保存当前的配置,当有异常发生时,如果属于系统能检测到的异常,会先保存系统配置,然后关闭进程。

4.2.2 二次开发接口的实现

业务开发工程师在知道二次开发接口,而不知道程序的主体的条件下,就可以通过重写二次开发接口迅速的将新业务集成到飞信机器人系统中,使软件的开发效率大幅度提升。

插件是实现二次开发接口的一个最优选择,使用插件技术的软件架构是一种很好的组件化程序设计方法。插件是一种遵循统一的预定义接口规范编写出来的程序,应用程序在运行时通过接口规范对插件进行调用,以扩展应用程序的功能。插件最有优势的地方在于其实现“运行时(run-time)”功能扩展。这意味着软件开发者可以通过公布插件的预定义接口规范,从而允许第三方的软件开发者通过开发插件对软件的功能进行扩展,而无需对整个程序代码进行重新编译。运行时的软件功能扩展其优点是显而易见的。

(1) 对软件开发者而言,只需对主程序和某些常用插件进行更新和维护,然后通过公布插件接口吸引第三方的软件开发者对主程序的功能进行扩展。

(2) 对最终用户而言,可以通过有选择地购买第三方提供的插件实现自己所需要的功能,从而实现最佳性价比组合,以节省不必要的开支。

在.NET平台下使用插件技术,其基本原理是利用.NET平台的反射技术。反射使用Assembly定义和加载程序集,加载在程序集清单中列出的模块,以及从此程序集中查找类型并创建该类型的实例,将类型绑定到现有对象,或从现有对象获取类型并调用其方法或访问其字段或属性。其基本原理在2.2.1.3小节已经讨论过。

实现动态加载程序集的方法有两个:

(1) 把可能发生变化的部分提取出来,并封装成程序集文件。

(2) 动态加载程序集,根据配置文件,动态调用合适程序集,来完成用户的功能需求。如果用户需要改变或增加新功能,只需更改配置文件,提供新的程序集。

在具体实现中,每个插件都要实现一个基类,即从这个抽象类继承新的类,在不同的插件中实现这个接口。当用基类的指针调用同一个接口时,由运行时

的多态性,在不同的插件中就可以表现出不同的行为。在主程序中,.NET 的反射技术使得插件被加载在程序集清单中,程序集中查找类型并创建该类型的实例,并将类型绑定到现有对象。有了这两个条件,插件的实现条件就具备了。

在插件的实现中,功能和接口的规划是设计的关键,因为插件通过一个约定的接口与主程序进行交互,接口需要根据需求来定义。一个合理和统一的接口函数,对插件以后的扩展和维护非常重要,如果插件的接口定义合适,将来业务的扩展将会受益匪浅。

在本课题的实现中,抽象出的插件接口如下所示。

```
public interface IConfigurableStateBase
{
    bool UserStateProcessMsg(UserInfo currentUser, string
        currentInput, string UserStateDescription, out string
        processResult);//非初始化状态处理函数
    string GetStateHelloMessage(UserInfo currentUser, string
        stateName,string inputMenu);//用户刚打开对话框的处理函数
    void UploadUserState();//加载用户状态。
    void DayPassed(Object source, System.Timers.ElapsedEventArgs
        e);//一天结束后的处理函数。
    void SaveSystemInfo();//保存系统信息函数。
}
```

所有的插件要实现上述接口。插件接口所有的成员函数都是在进行详细需求分析后设计的,将飞信用户和飞信机器人对话的状态,过程等尽可能多的信息抽象出来,放在插件接口中,来支持未来的可扩展性。实际应用证明该接口能够有效的支持不断变化的扩展应用。

4.2.3 智能对话的实现

飞信机器人作为飞信用户的联系人,两者之间通过发送飞信文本消息的方式交流,飞信机器人主要为飞信用户提供各种信息类的服务。飞信用户在打开与飞信机器人对话的窗口后,飞信机器人检测到飞信用户的动作,主动向飞信用户推送能够提供的信息服务,以菜单的方式组织。飞信用户可以用两种方式与飞信机器人对话,一种按照飞信机器人的提示,通过发送菜单相应级别的编号进入相应的菜单,另一种方式是直接输入想要查询的内容。飞信机器人的处理流程如下图所示。

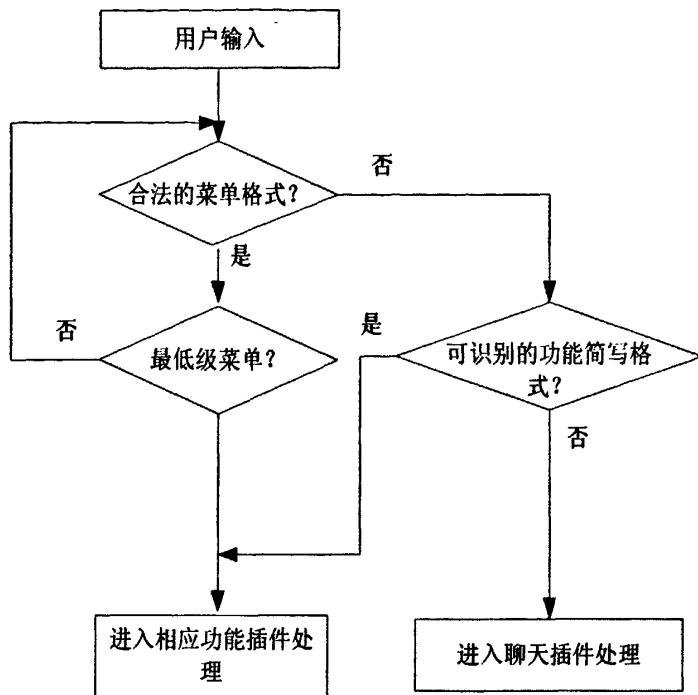


图 4-6 智能对话的处理流程图

在飞信机器人的配置文件中，以菜单的方式来配置每一级给飞信用户提供的界面以及需要加载的功能插件，在启动系统后，只会动态加载配置的插件，当飞信用户初始化和飞信机器人的对话窗口后，飞信机器人会以菜单的方式主动推送飞信机器人提供的服务，这是一种简洁，直观的方式，可以使飞信用户迅速了解到飞信机器人能够做的事情。在本课题的实现中，功能是按照菜单组织的，每一级菜单可以配置自己的名称简写。用户可以通过输入常规的菜单编号来进入相应的菜单级别，也可以直接输入菜单名称的简写，或者系统能够识别的输入，来直接进入相应菜单，或者获取相应服务。当用户的输入使系统无法判定是自己能够提供的服务时，会直接进入聊天插件，和用户进行属于娱乐性质的聊天。本课题的实现方式能够清晰明白的将飞信机器人能够提供的信息服务展示给飞信用户，飞信用户也可以很容易的通过提示获取相应的服务。

飞信机器人与飞信用户的交互，主要依赖于文字的沟通，在保证飞信机器人系统能够以简洁易懂的界面展示给飞信用户，飞信用户能够按照提示完成和飞信机器人的信息交互，从而获得其需要的服务的基础上，还应当使飞信用户对飞信机器人的满意度增加。人工智能是研究使计算机完成人类智能任务的学科，自然语言理解是人工智能领域中的核心课题之一，本课题已经完成了简单的语义理解，随着飞信机器人系统支持的业务增多，用户的使用人数增多，对用户的行为进行数据挖掘，加强对人工智能

方面的研究是下一步的目标。

4.3 系统实际运行情况

飞信机器人系统目前已经应用在上海, 江苏, 陕西等十余个省市移动官方机器人上, 飞信用户只要添加官方机器人飞信号为好友, 就可以通过和飞信机器人的对话, 来查询新闻、天气预报、股票、基金等信息。用户还可以通过飞信机器人来了解或参加移动面向移动用户的很多活动。除此之外, 越来越多的增值业务也被飞信机器人系统所支持, 用户可以直接通过飞信机器人来下载彩铃等, 免去了搜索网页的麻烦。

下图是飞信机器人系统的一个典型应用。

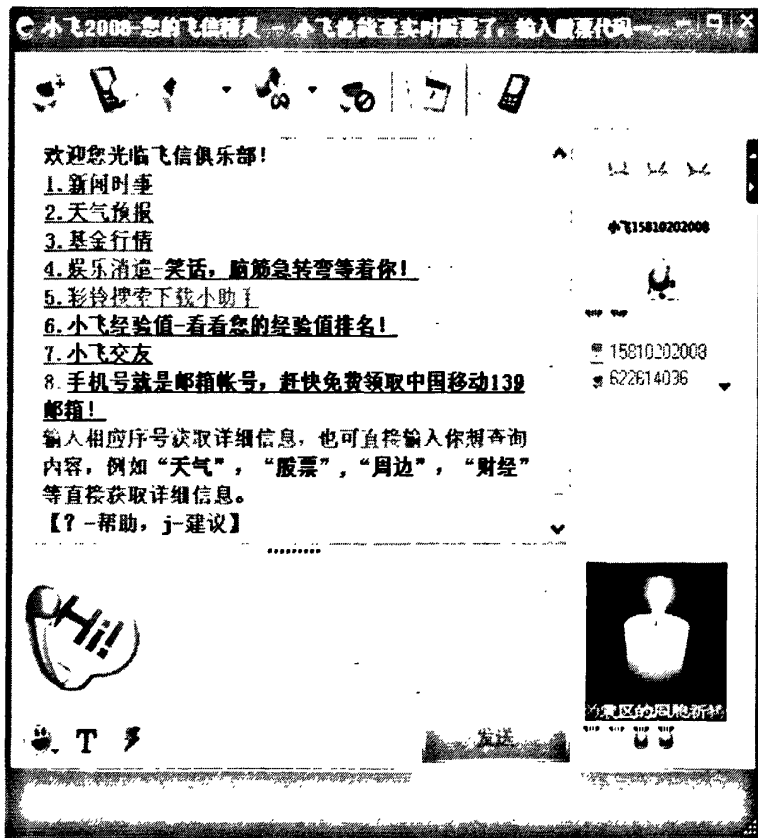


图 4-7 飞信机器人展示界面

如图 4-7 所示, 当飞信用户向小飞 2008 发出添加好友请求后, 小飞 2008 自动将飞信用户加为好友, 当飞信用户打开与小飞 2008 的对话窗口时, 小飞 2008 主动推送图 4-7 展示的内容给飞信用户。这时飞信用户可以按照小飞 2008 发送过来的内容提示, 发送数字 1-8 进入相应级别的菜单, 也可以输入“天气”,

“股票”等关键字，直接获取所需的服务。目前飞信机器人系统支持的业务达到百余种，策划人员可以通过修改配置文件，自定义显示刚给用户的样式以及支持的业务。

飞信机器人在推广过程中，按照设想的模式一步一步向前发展，在实际运营过程中，积累了大量的活跃用户，有很多用户对飞信机器人提供的某些他们感兴趣的服务（游戏，股票，基金等）产生了粘性。服务提供商对飞信机器人也产生了兴趣，他们寻求和飞信机器人合作的方式，在飞信机器人上添加新服务，可以通过重写二次开发接口，或者通过 Web Service，非常的快捷和方便。通常从确定新业务需求，到开发，测试，直到上线只需 3 天时间。比在飞信客户端直接引入一项新的业务，要快了数倍，更重要的是，与飞信机器人对话，类似与人之间的沟通，是一种互动的形式，在用户的角度，他们更愿意接受这种方式。

在实际运营过程中，可能会支撑一些并发用户数量非常大的业务，例如一个省的限时答题，短信或 PC 投票等，系统承受住了大流量的考验，经过大半年的运营，机器人平台累计好友数量达到数百万，累计对话次数超过 1000 万次，实际忙时单帐号最大处理流量超过 2000 次请求/分钟。

飞信机器人系统在飞信机器人上的应用现状和良好的发展趋势，使我们有理由相信飞信机器人是飞信杀手级的扩展应用，并且已经进入了向信息综合服务平台方向发展的轨道。

4.4 本章小结

在第三章系统设计的基础上，本章介绍系统关键部分的具体实现过程，并且介绍了系统在实际中的应用情况。

第五章 论文总结

2006年中国移动借助强大的网络平台和资源优势,推出了飞信IM,它是国内唯一能够跨平台使用的即时通讯工具,移动飞信在3年左右的时间里用户渗透率已经高达20.5%,占据市场第二的位置,已经超过了MSN。然而,从成长阶段来说,飞信还处在幼年期,没有较多的扩展应用来实现增值服务,在未来的发展模式上还要不断的探索。

飞信机器人是飞信比较有前景的一个扩展应用,本文设计的飞信机器人系统实现了在飞信即时通讯平台上搭建综合信息服务平台的任务。飞信用户可以添加飞信机器人为好友,用PC客户端或手机客户端和飞信IM对话,获得他们想要获取的服务。由于飞信机器人系统的设计使用了分层的架构,结合插件,多线程,Web Service等具体技术,使飞信业务的扩展非常容易。

在经过了详细的需求分析阶段,系统设计阶段,代码实现阶段后,飞信机器人系统得以顺利的实现,并且投入了实际应用中。在实际应用中,飞信机器人系统体现了良好的扩展性和稳定性。为了实现将飞信机器人系统变为综合信息平台的目标,采取了三步,将飞信机器人逐渐引入信息综合服务平台方向发展的轨道,

1. 和江苏,上海等省移动合作,飞信机器人系统使用官方飞信号,移动给予宣传。

2. 官方飞信号使飞信用户信任飞信机器人,飞信机器人为飞信用户提供信息类的服务(比如用户最关心的新闻,天气预报,游戏等),或者移动可以利用飞信机器人推出一些促销活动,让用户通过飞信机器人参加促销活动,提高飞信机器人的知名度,增加用户的粘性。在此过程中,开发了几十种插件,由于系统设计的时候考虑到未来发生的很多可能,使插件的开发变得非常高效。也充分体现了系统的可扩展性。

3. 当官方的飞信机器人已经吸引了大量用户的时候,有很多服务提供商希望和飞信机器人合作,共同开发增值业务。陆续有彩铃服务提供商,139邮箱等与飞信机器人系统进行了合作,飞信用户通过飞信机器人下载彩铃和访问139邮箱非常的方便和快捷,而对服务提供商来说,他们因此获得了大量的用户。飞信用户提出需求,服务提供商满足需求,飞信机器人作为载体,已经朝着信息综合服务平台的方向发展。

系统已经实际应用了大半年的时间,在此过程中承受住大流量的考验,机

机器人平台累计好友数量超过 100 万，累计对话次数超过 1000 万次，实际忙时单帐号最大处理流量超过 2000 次请求/分钟。实践证明，该系统有着较高的稳定性。

该系统还存在着一些不足，今后有以下几个重点研究方向：

1. 能够在系统用户持续增长，系统负载增大的情况下，使系统依旧保持稳定的特性。

2. 加强自然语言，人工智能方面的研究，最终能使用户以自然语言的方式向飞信机器人发出指令，实现真正人工智能的机器人。

3. 改进搜索的算法，在集成的业务越来越多的情况下，能够快速将用户的需求定位到合适的位置，缩短响应时间。

4. 优化数据库的性能，提高信息查询的速度。

5. 开发统一的 Web 管理平台，飞信机器人系统管理员将通过这个平台访问数据库，统一管理，维护飞信机器人。

参考文献

- [1] 中国互联网络信息中心 (CNNIC), 第 24 次中国互联网络发展状况统计报告, 2009 年 7 月。
- [2] 互联网络信息中心 (CNNIC), 中国即时通信用户调研报告, 2009 年 12 月。
- [3] Wu-Yuin Hwang, Sheng-Yi Wu, Hung-Cheng Chen, The effect of MSN Robot on learning community and achievement, International Conference on Digital Game and Intelligent Toy Enhanced Learning, 2008(18):207-208.
- [4] Ong Sing Goh, Chun Che Fung, Kok Wai Wong, VisualChat: A Visualization Tool for Human-Machine Interaction, International Conference on Web Intelligence and Intelligent Agent Technology, 2008(318):131-134.
- [5] Chin Chin Wong Hoh, S. Pang Leang Hiew, Value Proposition of Mobile Instant Messaging, Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference, 21-25 Sept. 2007 Page(s):3833 - 3836.
- [6] <http://www.alicebot.org/>.
- [7] M. Hatton, How to Successfully Implement Mobile IM, Yankee Group April 2006.
- [8] Wiral, "Wireless Instant Messaging: Market Opportunity and Business Models," Wiral White Paper September 3 2001.
- [9] D. Hugo, F. Pinto, F. Goncalves, J. Ferreira, J. Tefke, J. L. M. Terres, J. C. L. Calvet, P. Nieminen, and R. Rautavuori, "Mobile Instant Messaging - Technologies, Cost Analysis, and Business Potential," EURESCOM November 2003.
- [10] A. Herstad, B. Farshcian, B. Berntsen, G. Gylterud, and J. C. L. Calvet, "Business Analysis for IM Presence Services," EURESCOM November 2003.
- [11] G. Tatai, A. Csordas, A. Kiss, A. Szalo, and L. Laufer, "Happy Chatbot, Happy User," in Intelligent Virtual Agents, vol. 2792, 2003, pp. 5-12.
- [12] 付安民, 张玉清, 飞信即时通监控系统的设计与实现, 工程应用技术与实现, 229-231, 2008.7.
- [13] Brian Noyes, Data Binding with Windows Forms 2.0: Programming Smart Client Data Applications with .NET, Addison-Wesley Professional: 2006.
- [14] 王辉, 施小英, 中间件服务及其集成框架, 计算机工程与应用, 1999, 24(9):25-27.
- [15] 康博, C#.NET 网络程序设计, 科学出版社, 2004.
- [16] Andrew Parsons, Nick Randolph, Visual Studio 2005 高级编程, 清华大学出版社, 2008.
- [17] 威尔, .NET 编程先锋, 中国青年出版社, 2001.

- [18] 巴纳比, .NET 分布式编程:C#篇, 清华大学出版社, 2004。
- [19] 雷诺兹, .NET 企业应用高级编程.C#编程篇, 清华大学出版社, 2002。
- [20] 费拉基亚蒂, .NET 数据服务 C#高级编程, 清华大学出版社, 2002。
- [21] Thomas Erl, SOA 概念、技术与设计, 机械工业出版社, 2007。
- [22] Sandy Carter, SOA & Web 2.0:新商业语言, 清华大学出版社, 2007。
- [23] Thomas Erl, SOA 服务设计原则, 人民邮电出版社, 2009。
- [24] 吴健, UML 基础与 Rose 建模案例, 人民邮电出版社, 2007。
- [25] Hansen K, Damm C. Instant collaboration: Using Context Aware Instant Messaging for Session Management in Distributed Collaboration Tools, Proceedings of the second Nordic conference on Human Computer Interaction, N Y 2002(NordiCHI 10/02):279-282.
- [26] Donghui Feng, Erin Shaw, Jihie Kim, Eduard Hovy, An intelligent discussion-bot for answering student queries in threaded discussions, ACM Press, New York NY USA, 171-177.
- [27] Winograd, T. 1987. A Language/Action Perspective on the Design of Cooperative Work. Human-Computer Interactions, 3:1, pp. 3-30.
- [28] Cameron, A. F., & Webster J., Unintended consequences of emerging communication technologies, Instant messaging in the workplace. Computers in Human Behavior, 12(1), 85-103, 2003.
- [29] 宋爱, Web 挖掘研究综述, 计算机科学, 2002-12。
- [30] 王大震, 数据网格中的数据管理服务框架, 分析计算机工程与设计, 2004-12。
- [31] 曾春, 邢春晓, 个性化服务技术综述, 软件学报, 2002-10。
- [32] 孙增忻, 张在兴, 智能控制理论与技术, 清华大学出版社, 1997-07。
- [33] Xin Chen, 应用框架的设计与实现——.NET 平台, 电子工业出版社, 2005。
- [34] 朱德利, 数据挖掘与商业智能完全解决方案, 电子工业出版社, 2007-10。
- [35] G. Tatai, A. Csordas, A. Kiss, A. Szalo, in Intelligent Virtual Agents, vol. 2792, 2003, pp. 5-12.
- [36] O.S.Goh, C.C. Fung, and K.W.Wong, Query Based Intelligent Web Interaction with Real World Knowledge, New Generation Computing, vol.26, pp.3-22,2008.
- [37] C.Torrey, A.Powers, S.Fussell, Exploring Adaptive Dialogue Based on a Robot's Awareness of Human Gaze and Task Progress.in Human-Robot Interaction (HRI'07), 2007。
- [38] Polidoro, A. Salsano, S., Bartolomeo, Simple Mobile Services for IMS. Next Generation Mobile Applications, Services and Technologies, NGMAST '08. The Second International Conference on, 2008.
- [39] RFC 3261, SIP: Session Initiation Protocol, June 2002.

致谢

值此论文完成之际，我首先要感谢我的导师郭志刚教授，寿国础教授。两位老师渊博的知识，对学科前沿敏锐的洞察力，使我在整个硕士学习期间受益匪浅。两位老师严肃的科学态度，严谨的治学精神，精益求精的工作作风，深深地感染和激励着我。从课题的选择到项目的最终完成，两位老师都始终给予我细心的指导和不懈的支持。在此谨向两位老师致以诚挚的谢意和崇高的敬意。

感谢胡怡红老师在我成长中所给予大量的、无私的指导和帮助、老师的责任感、计划性以及执行力，都值得我去学习。

我要感谢在项目过程中，企业导师孙琛先生，在项目上给予我的锻炼机会，和对我的信任。在这一年多的时间，使我伴随着“飞信”项目成长。

在此，我还要感谢在一起愉快的度过研究生生活的学八 908 宿舍各位同学们，正是由于你们的帮助和支持，我才能克服一个一个的困难和疑惑，直至本文的顺利完成。

在论文即将完成之际，我的心情无法平静，从开始进入课题到论文的顺利完成，有多少可敬的师长、同学、朋友给了我无言的帮助，在这里请接受我诚挚的谢意！

最后我还要感谢培养我长大含辛茹苦的父母，谢谢你们！

攻读学位期间发表的论文

- [1] 杜娟. 一种支持有效移动性的移动 ipv6 和 sip 的集成架构. 中国科技论文在线[J]. 2009 年 11 月

