# 中山大学硕士学位论文

# 基于可视化检索的<br>广告信息增强系统的设计与实现

Design and Implementation of Visual Search-based Advertising Information Augmentation System

学位申请人：刘晓慧

导师姓名及职称：朝红阳教授

专业名称：软件工程

院、系（所）：软件学院

论文答辩委员会主席：_____

委员：_____

_____

_____

_____

二零一三年五月

## 论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

## 学位论文使用授权声明

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅，有权将学位论文的内容编入有关数据库进行检索，可以采用复印、缩印或其他方法保存学位论文。

保密论文保密期满后，适用本声明。

学位论文作者签名：　　导师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

论文题目： 基于可视化检索的广告信息增强系统的设计与实现

专业： 软件工程

硕士生： 刘晓慧

指导教师： 朝红阳（教授）

# 摘 要

本论文主要来源于微软亚洲研究院与中山大学智能信息处理和云计算实验室的合作项目：广告增强系统工程。本文在实验室和中国电信广州研究院合作的分布式图像检索平台 iSimilar 基础上，利用已有的核心技术及架构，结合分布式计算、网络爬虫、图像检索、手机应用开发等技术，为此项目提出了解决方案。同时设计和实现了一个端到端的在线移动广告信息增强系统——iSearch。该系统向后端广告发布商提供广告注册服务，向前端用户，特别是移动设备用户提供广告识别服务。

本文的主要工作有以下几个方面：（1）设计并实现了一个基于可视化检索的广告信息增强系统；系统的主要模块有：①广告注册模块，包括用户注册、登录、退出，广告信息上传，用户个人信息管理及广告信息管理等功能；②可视化检索模块，提供了电影、衣服、广告三个频道的信息检索功能。③手机客户端，提供了在 Android 及 Windows Phone 智能手机上使用本系统可视化检索功能的方法；④数据操作模块，包括数据抓取、分析、处理及模板管理等功能。（2）对开源网络爬虫 Heritrix 和 Nutch 进行了实验和分析；并在 Nutch 的基础上实现了一个可定制的分布式数据抓取工具，该工具适用于对有相同结构的网页进行准确的信息抓取。（3）封装了一个 HTTP 接口，客户端可方便地通过 HTTP 协议获取系统的图像检索服务。

本文的主要贡献是改进了一些和项目相关的现有技术：主要是提出了一种基于 XPATH 的模板信息提取方法，实现了对网页指定数据的准确提取；并将该方法与网络爬虫结合，一定程度上解决了现有网络爬虫不能进行数据的准确抓取的问题。此外，本论文改进了 iSimilar 平台的索引方案，解决了原方案不支持新数据实时搜索的问题。本论文提出了一种创新的广告信息增强系统的应用模式；利用图像检索技术及移动互联网，提供了一个端到端的可视化移动搜索平台，人们几乎可以随时随地获取他们感兴趣海报广告的更多相关信息，达到有效增强海报广告效果的目的。

本论文根据软件工程的开发流程，对 iSearch 系统进行了详细的分析、设计以及实现。目前，iSearch 系统各模块的功能已经实现，可以通过 Android 客户端、Windows Phone 客户端及浏览器简单快速地使用系统提供的检索功能。分布式数据抓取工具已经被验证能够准确提取指定信息，并完成了对“时光网”的指定数据的准确抓取。系统的有效代码约一万五千行。

**关键词：**分布式，爬虫，信息提取方法，移动搜索

Title: Design and Implementation of Visual Search-based Advertising Information Augmentation System

Major: Software Engineering

Name: Xiaohui Liu

Supervisor: Hongyang Chao (Professor)

# Abstract

The research subject of this paper mainly came from Microsoft Research Asia and Sun Yat-sen intelligent information processing and cloud computing laboratory's cooperation project: Advertising enhance system engineering. On the basis of the laboratory and China Telecom Guangzhou Research Institute's distributed image retrival platform, iSimilar, this paper used the existed core technology and architecture, combined with distributed computing, Web crawler, image retrieval, mobile application development technoloty, proposed a solution for the project. This paper also designed and implemented an End-to-End Online Mobile Advertising Information Augmentation System – iSearch. This system provided advertising registration services to the back-end publishers, and advertising recognition services to front-end users.

The main work of this paper includes the following aspects: (1) Designed and implemented a visual search-based advertising information augmentation system; major modules are: ① Advertising registration module, provides the following functions: user registration, user login, advertising information uploads, personal information management, advertising information management, and so on. ② Visualization retrieval module, provides information retrieval function for three channels(movies, books, advertisements). ③ Mobile client, providing a way to use the visual search function on Android and Windows Phone smartphone. (2) Did some experiments and analysis on open source web crawler Hertrix and Nutch; and implemented a custom distributed data crawler based on Nutch. This tool was applicable to crawl accurate information from the web pages with same structure. (3) Encapsulated an HTTP interface, the clients could obtain system's image retrieval service

through the HTTP protocol easily.

The main contribution of this paper was to improve some project relevant prior art: we proposed a XPATH-based template information extraction method, archieve the accurate extraction of the specified data from web page. This method in conjunction with the web crawler, to a certain extent solves the problem of existing web crawler can not captured data accurately. In addition, thist paper improved the indexing scheme of the iSimilar platform, solved the problem of the original scheme doesn't support the new data real-time search.

This paper proposed an innovative advertising information augmentation system application mode; provide an end-to-end visual mobile search platform with image retrieval technologies and mobile Internet; people can get more related information about the posters advertising they are interested in almost anytime, anywhere, to effectively enhance the poster advertising effectiveness.

Detailed analysis, design and implementation of iSearch system according to the development process of software engineering was made in this paper. Currently, each module's function of iSearch system has been archieved, people can use the search function provided by the system through the Android client, Windows Phone client and Web browser. The distributed data crawler has been verified to be able to accurately extract specified information, and completed the accurate crawl specified data of "Mtime". Effietive code of the system is about 15,000 lines.

**Key Words:** Distributed; Crawler; Information extraction method; Mobile search

# 目录

**第一章　前言** ..... **1**
1.1　项目背景和意义 ..... 1
1.2　研究发展现状 ..... 2
1.3　论文的主要工作与贡献 ..... 8
1.4　论文结构 ..... 10
**第二章　ISEARCH 系统关键技术分析与介绍** ..... **11**
2.1　整体解决方案 ..... 11
2.2　HADOOP ..... 12
2.3　网络爬虫技术 ..... 14
2.4　网页信息抽取技术 ..... 19
2.5　增量索引方案 ..... 25
2.6　SSH 框架 ..... 27
2.7　AJAX 技术 ..... 27
**第三章　ISEARCH 系统需求分析** ..... **29**
3.1　ISEARCH 系统概述 ..... 29
3.2　ISEARCH 系统主要用例分析 ..... 30
3.3　ISEARCH 系统领域分析和建模 ..... 44
3.4　ISEARCH 系统的其他需求 ..... 46
**第四章　ISEARCH 系统架构设计** ..... **48**
4.1　ISEARCH 系统架构及原理 ..... 48
4.2　ISEARCH 系统重要业务用例实现 ..... 51
4.3　HTTP 接口概要设计 ..... 56
4.4　删除广告方法概要设计 ..... 57
4.5　数据库设计 ..... 58
4.6　ISEARCH 系统出错处理设计 ..... 60
**第五章　ISEARCH 系统模块设计** ..... **62**
5.1　ISEARCH 系统模块概述 ..... 62
5.2　数据操作模块 ..... 63
5.3　WEB 可视化检索模块设计 ..... 72
5.4　广告注册模块 ..... 73
5.5　手机客户端 ..... 75
**第六章　ISEARCH 系统部署与应用** ..... **76**
6.1　开发环境与运行环境 ..... 76
6.2　ISEARCH 系统测试 ..... 78
**第七章　总结与展望** ..... **94**
**参考文献** ..... **97**
**致　谢** ..... **99**

# 第一章　前言

在图像检索技术日渐成熟及移动互联网日渐普及的条件下，微软亚洲研究院提出了基于可视化检索的海报广告信息增强系统的项目，简称 iSearch。目前，国内外已经有不少基于内容的图像检索平台，如谷歌、微软、百度等，但都没有提供专门的广告搜索频道，并且搜索结果缺少相关的文字描述，不能很好地起到增强广告效果的作用。因此，本论文结合分布式计算、网络爬虫、图像检索、移动应用开发等技术，为 iSearch 提出了自己的解决方案。

本章主要介绍了项目的背景和意义，并通过对国内外发展现状的分析证明该系统的意义和可行性。此外，本章阐明了论文的主要工作，系统的设计与实现以此为基础展开。最后，对论文的基本结构进行了说明。

## 1.1　项目背景和意义

本论文主要来源于微软亚洲研究院与中山大学智能信息处理和云计算实验室的合作项目，该项目的主要目的是构建一个广告信息增强系统。本文为该项目提出了解决方案。

根据中华人民共和国国家工商行政管理总局公布的数据，2012 年全国的广告经营额已达 4698 亿元，比 2011 年增长 50%以上，并占 2012 年中国 GDP 的 0.9%[1]。目前，中国广告的市场总规模已超过德国跃居世界第三，仅次于美国和日本。随着广告投入的增加，广告投资商所期望获得的收益也相应提高。广告是通过刺激消费心理、引导消费行为来为广告投资商带来效益的。如何利用有限的资源和投入资金来尽可能地增强广告的效果，成为越来越多广告投资商所关注的问题。J.A. García *等人曾针对广告图像的视觉效果进行了一系列实验，结果表明能够简洁明了地传递信息的广告图像能取得更好的广告效果[2]。这是从广告本身的设计出发来增强广告的吸引力，但人们能获取的信息仍然限于广告本身。特别是对于海报广告来说，它所能传播的信息更是为交互性及篇幅等条件所限制。

根据第 31 次《中国互联网络发展状况统计报告》，截至 2012 年底，我国搜索引擎用户规模为4.51亿，占网民总数的80%[3]，搜索引擎已经成为网民获取信息的重要途径，其中也包括了广告信息。对于广告图像而言，使用基于文本的搜索引擎来进行检索，由于对图像的描述、标注方式及语义等因素的影响，可能致使搜索结果与预期有很大差异；使用图像作为关键字进行检索可以更直观地描述搜索条件。目前，谷歌、百度、微软 Bing 等都推出了基于内容的图像检索服务，但它们都没有专门的海报广告搜索频道，并且搜索的结果缺少相应的标注信息，不能很好地起到广告增强的作用。

本文结合网络爬虫、基于内容的图像检索、分布式计算、手机应用开发等技术，构建了一个提供分布式搜索服务的在线广告信息增强原型系统——iSearch。这是一种创新的广告信息增强系统应用模式，为潜在消费者提供了方便快捷获取感兴趣商品信息的途径，以达到有效增强海报广告交互能力及宣传效果的目的。此外，针对现有网络爬虫无法对网页信息进行分析和提取的问题，本文提出了一种基于 XPATH 的模板信息提取方法。该方法可以准确定位要提取的信息的结点，达到准确提取信息的目的。将网络爬虫技术与信息提取技术结合，可实现快速准确抓取数据的目的。

## 1.2 研究发展现状

随着社会经济和现代科学技术的高速发展，广告业也随之迅猛发展。如何提高广告的效果越来越为广告投资商所重视。搜索引擎作为当前人们获取信息的重要途径，通过广告图像搜索获取广告信息可以有效地增强广告效果；但目前已有的图像检索平台由于标注信息缺乏，针对性不强等因素，都不能很好地满足广告增强的需求。对于搜索引擎来说，数据源是必不可少的部分，对于 iSearch 系统来说，需要大量有标注的图像作为数据基础；而目前用于快速搜集数据的网络爬虫技术不能实现对网页数据进行准确提取。综合来看，当前的应用及技术并不能很好地满足 iSearch 系统的需求。下文对广告效果增强研究，国内外的图像检索平台，目前比较常用的开源网络爬虫，网页信息提取技术，以及分布式系统架构进行简单介绍。

### 1.2.1 广告效果增强的研究

随着科学技术的迅猛发展，现代广告业得到了前所未有的发展，广告的形式和种类越来越多，有电视广告、霓虹灯广告、橱窗广告、报刊广告、移动公交广告、植入式广告、互联网广告等。

海报广告又叫招贴广告，距今已有一百多年历史。海报广告的简洁、经济实用、易传播等特点使它仍是现在最流行的广告形式之一。然而缺乏交互性及内容延展空间等因素的限制使得海报广告的效果为之减弱。目前，已有不少学者开始从改善海报交互性的角度来研究增强海报广告的效果。

近年来，移动互联网发展迅速。图 1-1 中的数据表明，至 2012 年 12 月，我国的手机网民规模已经达到 4.2 亿，年增长率达 18.1%，占网民总数的比例也由 69.3%上升到 74.5%。随着移动互联网的日渐普及和手机使用率的提高，手机广告越来越多地出现在人们的视线中。手机广告的个性化，互动性，实时性等特点使其越来越受广告投资商的青睐。随着手机广告的兴起，对手机广告效果的研究也越来越受关注。顾其威等人根据用户对广告的访问历史和操作模式等建立用户综合兴趣模型，并验证了模型在手机广告推荐中的有效性和优越性[4]。Jung woo Lee 等人则是从记忆、心理和消费行为三个方面对推式手机广告效果进行了实验分析，结果表明多媒体广告和优惠广告与其它类型广告取得的效果不同，并且人们的负面看法会对广告效果产生影响[5]。

图 1-1：中国手机网民规模及其占网民比例[1]

与前文提到的增强广告效果的方法不同，在本文提出的广告信息增强系统中，用户是广告信息的主动接收方，系统根据用户的实际请求向其返回相应的广告信息，是对用户自身感兴趣广告的信息扩展。

### 1.2.2 图像检索平台

自图像检索出现以来，它经历了两大阶段：（1）20 世纪 70 年代，基于文本的图像检索（Text-based Image Retrieval，简称 TBIR）产生，它存在着两大难题：①对图像进行手工标注的工作量太大；②手工标注会受到人的主观影响并且由于语义、描述方式等差异而导致的不精确性[6]。（2）基于内容的图像检索（Content-based Image Retrieval，简称 CBIR）[7]出现于 20 世纪 90 年代，它以图像作为搜索字，利用图像的内容语义特征来进行检索，一定程度上消除了 TBIR 由于文本标注等因素导致的搜索结果偏差。

经过多年的发展，图像检索技术在国内外取得了不少成就，现在已经有很多基于内容图像检索的原型系统，如：

（1）QBIC 系统[8]：该系统由 IBM 开发，使用色彩、形状、纹理和手绘草图这些图像内容作为基础进行检索，有图像入库、特征计算和图像查询三个逻辑步骤。QBIC 基于内容的图像检索技术已制成独立产品，如 IBM 数字图书馆、DB2 数据库的图像扩展等工具软件[9]。

（2）Virage 系统[10]：该系统由 Virage 公司开发，它向开发人员提供了一个“插入原语”来解决具体的图像管理问题的开放框架。Virage 引擎提供的基础设施可以用来解决高层次的问题，如自动、无监督的关键字分配，或图像分类。

（3）VisualSEEK 系统[11]：该系统由哥伦比亚大学研发。它提供了通过色彩、纹理和空间布局的图像查询工具。该系统目前已在电子图书馆等领域得到了应用[9]。

不少学者提出了各自的图像检索改进算法，并取得了一定的效果。Mahmood R. [12]等人在2009年提出了一个使用内核机器和选择性采样的相关反馈自适应图像检索系统，并验证了该系统提出的方法召回率高于多轮查询反馈学习算法[13]及径向基函数[14]。在同

[1] 本图来源于第 31 次《中国互联网络发展状况统计报告》

一年，David Picard[15]等人提出群体智能在分布式图像检索上的应用，该应用通过主机上的标记来检索图像，并且根据先前搜索会话的相关反馈来改善检索结果。而 Gwénolé Quellec[16]等人在 2010 年提出自适应非分离的小波变换及它在基于内容的图像检索上的应用，实验结果表明自适应非分离的小波变换方法比自适应分离的小波变换方法及二元树复小波变换方法所取得的检索精度更高。

此外，图像检索系统也越来越多地出现在人们的现实生活中。如“图片谷歌”、“百度识图”、“搜狗识图搜索”、“淘淘搜”等基于内容的可视化图像检索服务已投入使用。

iSimilar 也是日渐成熟的图像检索技术的一个实际应用项目，它使用 SIFT[17]、GIST[18]等算法进行图像特征提取，并使用 LSH[19]算法对图像特征进行快速聚类；此外，iSimilar 对 LSH 算法进行了改进，以支持在线聚类。

在上文提到的众多基于内容的图像检索系统中，均没有以增强广告效果为目的而构建的系统或频道，检索结果缺少相关的标注信息，并且会出现检索结果的图片来源网页内容与检索图片无关的情况。现有的图像检索系统并不能很好地满足广告信息增强的目的。而 iSimilar 是一个分布式可扩展的开源图像检索平台，具有可通过配置方便地使用各种图像检索方法的优点，本论文将以 iSimilar 的核心架构为基础构建 iSearch 系统。

### 1.2.3 开源网络爬虫

搜索引擎必须有大量的数据做支撑，对于图像检索平台来说，图片源必不可少。随着网络的发展及网民数量的增加，互联网上的数据量呈几何级数增长，互联网成为主要的数据来源。网络爬虫技术的出现很大程度上满足了快速获取大量数据的需求。

网络爬虫最早出现于 1993 年，由麻省理工学院的 Mattew K Gray 用 Perl 编写，名为“World Wide Web Wanderer”，最初建立的目的是为了统计互联网上的主机数目[20]。

目前的网络抓虫技术已臻成熟，常用的开源爬虫项目有：

（1） Heritrix[21]

Heritrix 是 Internet Archive 的开源、可扩展、网络规模的归档网络爬虫项目，编程语言为 JAVA。该项目始于 2003 年初，目前的最新版本是 3.1.1，于 2012 年 5 月正式发布。

（2） Nutch[22]

Nutch 是 Apache 的子项目之一，是一个用 JAVA 语言实现的开源分布式搜索引擎框架。目前 Nutch 有 1.x 和 2.x 两个分支版本，它们之间的主要区别有：①1.x 版本将下载的数据存放在 HDFS[23][24]上；2.x 版本还支持存放在数据库中，如 HBASE[23]。②对不同类型的文档，1.x 版本使用不同的插件来解析；2.x 版本则是主要使用 Tika[25]来解析，解析插件数量减少。③2.x 版本将 Nutch 中的 URL 过滤、索引去重等网络爬虫的公共功能提取出来，可供其它爬虫使用。④去掉了 Nutch 1.x 中的索引/搜索功能，仅提供抽象层以整合其它索引/搜索功能。1.x 版本的性能相对 2.x 版本更稳定，并且对不同类型文件的解析方法的可配置性更好。目前 1.x 的最新版本是 1.6，于 2012 年 12 月 6 日正式发布；而 2.x 的最新版本是 2.1，于 2012 年 10 月 5 日正式发布。

现在，开源网络爬虫项目越来越多，但它们大多是提供网页的全文下载或索引，无法直接应用来满足 iSearch 系统需要大量图片及准确标注信息的需求，需要引入网页信息提取技术来实现对网页数据的分析和提取。Heritrix 及 Nutch 的对比分析将在 2.3 小节中进行阐述。

### 1.2.4 网页信息提取技术

网页信息提取即从网页中提取所需的内容，如标题、网页正文等。Mohammed Kayed[27]等人将网络信息抽取系统分为四类：（1）手动构造的信息提取系统，用户为每个 Web 站点编写包装器，这要求用户有大量的计算机和编程背景知识；这类系统的典型代表有 TSIMMIS、W4F、XWRAP 等。（2）有监督的包装器归纳系统，由用户提供已标记的初始训练集；WHISK、NoDoSE 等都属于这一类。（3）半监督的信息提取系统，包括 IEPAD、OLERA 和 Thresher；IEPAD 不需要已标记的训练页面，但要求用户指明要提取的内容及目标模式；OLERA 和 Thresher 则需要用户提供粗略的实例以生成提取规则。（4）无监督的信息提取系统，包装器的生成不使用任何已标记的训练实例且没有任何的用户交互；RoadRunner、EXALG、DEPTA 都是无监督的信息提取系统。

此外，刘秉[28]等人提出了基于树的局部调整的网页结构化数据提取方法，该方法分为两个步骤：（1）识别在一个页面中的个人数据记录，（2）从识别的数据记录中调整并提取数据项；实验结果表明，这种方法可以准确地对数据记录进行分段并从中提取数据，能够有效地从网页中提取感兴趣的信息。

根据 iSearch 系统的实际需求，需要对多种结构的网页进行数据分析和提取；而前文中提到的工具和方法都不能满足同时对多种类别和结构网页进行准确的信息提取的要求。为此，本文对上述问题提出了解决方案，将在 2.4 小节中进行介绍。

### 1.2.5 分布式计算

随着 Internet 的高速发展，互联网上的各类数据越来越多。可视化检索相对于文本的图像检索，计算量骤增。传统算法上的改进已经难以解决图像存储、计算及数据传递等一系列问题。分布式计算的出现使得可视化检索有了新的发展。

分布式计算主要研究如何把一个需要非常巨大计算量才能解决的问题分成许多小的部分，分配给联网的计算机进行处理，最后把这些计算结果综合起来得到最终的结果。目前，分布式计算被广泛应用于各个领域，如：SETI@Home[2]，这是一项利用全球联网的计算机共同搜寻地外文明的科学实验计划，截止至 2005 年 7 月 14 日已有接近 20 亿台计算机加入到了项目中；fightAIDS@home[3]，这是由非盈利的 Scripps 研究所主办的帮助设计爱滋病新药物的生命科学项目，在 2003 年 5 月 21 日结束的阶段 I 中，近 6 万台计算机完成了原本需要 1,400CPU 计算年才能完成 900 万项任务。

2005 年，Apache 正式引入 Hadoop[23]项目。Hadoop 是一个分布式系统基础架构，用户可以在不了解分布式底层细节的情况下，开发分布式程序，充分利用集群的高速运算和存储。Hadoop 为用户提供了一个能够轻松架构和使用的分布式计算平台，用户可以很容易地利用 MapReduce[23][26]框架在 Hadoop 上开发和运行处理海量数据的应用程序。另外，Hadoop 实现了一个分布式文件系统，简称 HDFS[23][24]，提供了可扩容的大容量存储能力。HBase[23]则是 Hadoop 上的一个高性能、面向列、可伸缩的分布式数据库，以 HDFS 作为其文件存储系统。

Hadoop 框架可以很好地解决可视化图像检索中遇到的大计算量和大容量存储问题。

[2] SETI 项目介绍：http://setiathome.berkeley.edu/

[3] fightAIDS 项目介绍：http://fightaidsathome.scripps.edu/

## 1.3 论文的主要工作与贡献

本论文的主要工作是设计并实现了一个基于可视化检索的旨在提高海报广告效果的广告信息增强系统，该系统主要具备以下功能模块：

（1） Web 端可视化检索模块。该模块以 iSimilar 检索平台的图像检索功能为基础，提供电影、衣服、广告三个频道的可视化检索功能，方便用户获取相关详细信息。

（2） 广告注册模块。该模块包含了用户管理功能，包括用户的注册、登录、退出系统、信息修改及管理上传广告的功能。已注册的用户登陆系统后可以使用该模块的功能来向系统提交海报广告图片及图片相关的扩展信息。

（3） 手机客户端。系统提供了 Android 及 Windows Phone 两个智能手机平台的可视化移动检索功能。用户在手机上安装相应的客户端之后即可使用系统提供的可视化检索服务。

（4） 数据操作模块。该模块主要包括四个部分：分布式数据抓取，通过定制的模板准确抓取相应的网站数据；数据分析处理，将抓取的分层数据进行分析并合并成完整的可供使用的单元数据；数据统计，统计抓取的图片及电影数量；模板管理，对为抽取不同网站信息所定制的模板进行统一管理。

在实现该系统的过程中，遇到的问题有以下几点：

（1） iSearch 系统检索服务需要大量图片及标注信息为基础，现有的网络爬虫技术大多只提供网页内容的全文下载或索引，缺少对信息的准确抓取方法，不能直接应用于 iSearch 系统。

（2） 对一事物的相关信息可能分布在不同网页中，如 Mtime 时光网的电影海报与电影介绍、电影影评皆不在同一页面中。从这些网页中提取出来数据也会被独立存储，若将这些数据直接应用在 iSearch 系统中，会出现图片标注信息不完整、遗漏等问题。因此，如何将分布在不同页面中的相关信息进行合并是系统实现过程中遇到的难点之一。

（3） iSearch 系统所使用的数据源中，图片的标注信息过长过多，不适合存储在 iSimilar 平台的图片数据库中。

（4） iSearch 系统使用了 iSimilar 平台提供的图片存储及基于内容的图像检索功

能。根据 iSimilar 平台的索引方法，向图片库插入新的数据时，需要手工对图片库内所有数据重构索引，才能检索到新添加的数据。这样的设计不能满足 iSearch 系统提供的广告上传功能需要支持较频繁的数据插入操作以及能够快速检索到新增数据的需求。

（5） iSearch 系统向注册用户提供了对已上传广告的删除功能，但目前 iSimilar 平台的图像数据库未提供数据删除功能，如何实现“伪删除”功能也是需要考虑的问题之一。

本论文的主要贡献是对以上提出的问题提出了解决方案，并取得了良好的效果，具体有以下几点：

（1） 提出了基于 XPATH 技术的可定制的模板提取方法，实现了准确提取所需数据的功能，弥补了现有爬虫只能做整页下载的不足。该方法利用 XPATH 路径表达式，准确定位到所需提取信息所在的网页结点，再结合字符串截取方法，实现了对文本数据及链接的准确提取。这个方法适用于具有相同结构的网页，可通过编写不同的模板来适应对不同结构网页的分析和信息提取。

（2） 对于分散的相关信息的分析与合并，要求在提取出来的信息中包含关联标记，通过在模板中添加除所需提取信息之外的关联标记提取为实现。例如：页面 1 是电影 A 的介绍信息，页面 2 是电影 A 的海报图片列表，且两个页面中都包含电影 A 的网址，那么可以将电影的网址作为关联标记加入提取结果；对提取结果进行分析时，拥有相同关联标记的两组数据可被判定为相关信息，并进行合并。该方法未能实现通用，对拥有不同层次及结构的网页，需要根据实际情况来定义关联标记。

（3） 对于 iSimilar 平台的图片数据库不适宜存储过长文本数据的情况，提出了 MySQL 与 iSimilar 图片数据库结合的存储方案。利用 MySQL 数据库来存储文本数据，iSimilar 图片数据库存储图片数据，两者之间通过图片的 ID 进行关联。为了适应这个存储方案，本论文还重新封装了获取 iSimilar 平台图像检索功能的 HTTP 接口，通过该接口可以方便地从各个终端使用 iSimilar 可视化检索功能。

（4） 利用 Lucene 技术实现了对新插入数据的增量索引功能，避免了对所有图片重构索引的操作，基本上实现了对新增数据的实时检索。

（5） 利用 MySQL 数据库记录删除的图片 ID，并根据图片 ID 将已删除的数据从检索结果中剔除，该方法将在 4.5 小节中做进一步介绍。该方法并未实现数据的真正删除，随着删除数据的增加可能会影响到返回给用户的检索结果，在以后的工作中需要对这个问题做进一步的探讨。

此外，本论文提出了一种创新的广告信息增强系统的应用模式；利用图像检索技术及移动互联网，提供了一个端到端的可视化移动搜索平台，人们几乎可以随时随地获取他们感兴趣海报广告的更多相关信息，达到有效增强海报广告效果的目的。

本论文设计并实现了 iSearch 系统，主要使用 JAVA 和 JSP 作为编程语言，有效代码约一万五千行。

## 1.4 论文结构

本文一共七章，其组织结构如下：

第一章是前言，主要介绍项目背景和意义，对国内外技术发展状况进行了简单的分析，并阐述了本论文的主要工作、遇到的问题及贡献。

第二章对系统实现所涉及的相关概念和技术进行了介绍。通过对开源爬虫 Heritrix 和 Nutch 的实验分析，选取更适用于本系统的基础爬虫。**在本章中，将对本文提出的基于 XPATH 的网页信息提取方法进行介绍，这是本文的主要技术贡献之一。**

第三章至第六章对系统的分析与设计进行具体讨论。

第三章主要对 iSearch 系统进行详细的需求分析，并对其主要用例和领域模型进行设计。

第四章根据系统的需求进行整体架构的设计，实现了关键用例，并简单介绍了系统的数据库设计及出错处理。

第五章对 iSearch 系统模块设计进行讨论。

第六章是系统部署与应用。在这章中，对系统的开发、运行环境进行了介绍，并给出了系统的测试效果。

第七章对本文的工作进行归纳和总结，提出系统的不足及改进方案。

# 第二章　iSearch 系统关键技术分析与介绍

在第一章中，作者对课题的背景、论文的主要工作及贡献等做了简单介绍。

在本章中，我们先在 2.1 小节介绍系统的整体解决方案，然后在后面的小节中对系统中所涉及的概念及关键技术进行介绍。2.2 小节对本文使用的分布式系统框架 Hadoop 进行了介绍。在 2.3 小节中，对 Heritrix 和 Nutch 进行了对比和实验，验证了 Nutch 更适合于 iSearch 系统的抓取需求。在 2.4 小节中，介绍了本文提出的基于 XPATH 技术的可定制网页信息模板提取方法，旨在与爬虫技术结构以完成网页数据自动提取功能，**这是本论文的技术贡献之一。**2.5 小节对 iSimilar 原有的索引方法及改进后的方案进行了简单对比。在 2.6 和 2.7 小节中则分别介绍了 SSH 框架及 Ajax 技术。

## 2.1　整体解决方案

iSearch 系统从总体上来说分为四大模块：

（1）Web 可视化检索模块，用户通过浏览器可以方便地使用系统提供的电影、服装、广告三个频道的可视化图像检索服务。

（2）广告注册模块，用户可以通过注册及登录系统来提交海报广告图片及相关的扩展信息，并对上传的数据进行管理。

（3）手机客户端，用户在 Android 或者 Windows Phone 智能手机上安装相应的应用程序即可使用系统提供的可视化检索服务。

（4）数据操作模块，该模块主要负责模板的定制，分布式数据抓取，及数据的分析、合并、存储及统计任务。

搜索引擎必须有大量的数据资源做支撑，数据获取是建立搜索引擎至关重要的一个环节。在现有条件支持下，为了实现数据的快速获取和处理，本系统使用 MapReduce 分布式计算框架进行实现（在 2.2 小节中介绍）。而目前从互联网上快速获取数据的主要工具是网络爬虫，通过对 Heritrix 和 Nutch 两个开源网络爬虫的对比分析，我们选取了 Nutch 作为本文的基础爬虫（在 2.3 小节中进行分析）。然而现有爬虫只能进行全文下载，不能满足 iSearch 需要大量图片及标注信息的需求，本论文提出了可定制的信息抽取方

法（在 2.4 小节中介绍）。

构建索引也是建立搜索引擎的一个重要步骤，本文采用 iSimilar 平台的核心架构中包含了建立图像索引的方案，然而这个方案并不能很好地支持频繁的数据插入操作，本文为此问题提出了改进方案（在 2.5 小节中介绍）。

系统与用户的交互界面设计所采用的技术在 2.6 至 2.8 小节中进行介绍。2.6 小节对本文实现的手机客户端所使用的技术进行了简单介绍。此外，本文采用了 SSH 框架来构建 Web 客户端（在 2.7 小节中介绍），并使用 Ajax 技术来改善用户的系统体验（在 2.8 小节中介绍）。

## 2.2 Hadoop

Hadoop[23]是由 Apache 基金会开发的一个分布式系统的基础架构，于 2005 年作为 Nutch 的一部分正式引入。用户可以在不了解分布式底层细节的情况下，充分利用 Hadoop 的高速运算和存储能力开发分布式程序。Hadoop 具有高可靠性、高扩展性、高效性、高容错性等诸多优点。

### 2.2.1 HDFS

Hadoop Distributed File System（简称 HDFS）[23][24]是 Hadoop 的最底层构成元素之一，是 Google File System 的开源实现。它是一个运行在通用硬件集群上的分布式文件系统，有高容错性，采用流式数据访问模式。HDFS 的设计更偏重于适合批量处理，而不是用户交互式的；它更强调数据的高吞吐量，而不是低延迟的数据访问。

HDFS 采用主/从（Master/slave）架构。如图 2-1 所示，HDFS 集群是由一组特定的节点构建的，包括一个 NameNode 和多个 DataNode。NameNode 是一个负责管理文件系统命名空间并调节客户端访问的主服务器，存储了所有 HDFS 的元数据。DataNode 则是数据节点，负责存储实际数据并进行管理。当一个 DataNode 失效时，NameNode 会检查所有需要复制的块并将它们复制到其他的 DataNode 上，这使得 HDFS 具有较高的容错性。但是，HDFS 也存在一个明显的缺点——NameNode 单点失效，NameNode 失效将导致整个 HDFS 集群无法使用。目前采用复制 NameNode 文件的方法来避免由于元

数据的损坏而导致集群的所有文件丢失，但 NameNode 的自动重启和切换另外的 NameNode 仍不支持。

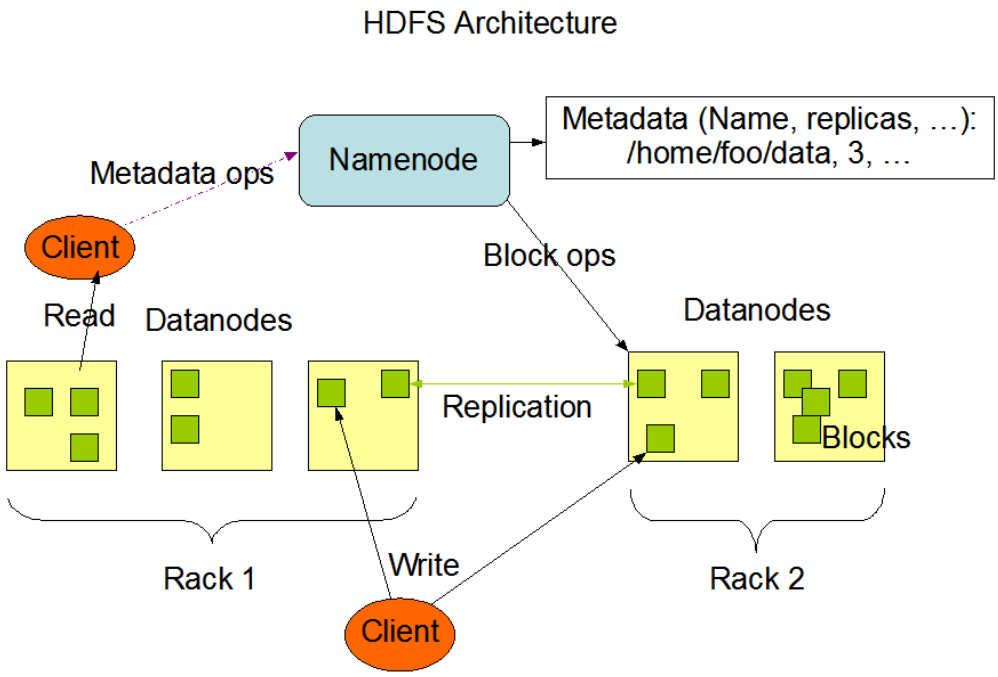



图 2-1：HDFS 架构[4]

## 2.2.2 MapReduce

Hadoop 的另一个核心部分是 MapReduce 编程模型[23][26]，是 Google MapReduce 的开源实现。Hadoop 可以运行用 Java、Ruby、Python、C++等多种编程语言编写的 MapReduce 程序。

Map 函数和 Reduce 函数是 MapReduce 编程模型的核心，它们的输入及输出均是 Key/Value 对格式。Map 函数和 Reduce 函数共同定义了 MapReduce 任务的行为，由用户来进行实现。Map 函数处理一个 Key/Value 对并生成一组中间的 Key/Value 对，Reduce 函数接收 Map 函数生成的中间 Key/Value 对，并将拥有相同的中间 Key 的值进行合并。使用 MapReduce 模型编写的程序会自动地在大型集群上并行化运行，这使得编程人员可以在没有任何分布式编程经验的情况下，也可以很容易地利用大型分布式系统的资源。

[4] 本图来源于《HDFS Architecture Guide》，参考文献[24]

Map/Reduce 任务的执行过程如图 2-2 所示，大致流程为：（1）MapReduce 库将输入文件分片，然后这个进程被拷贝到集群的其它机器上。（2）这些进程的副本有一个被称为 Master，其余称为 worker。Master 负责调度，为空闲的 worker 分配 Map 或 Reduce 作业。（3）被分配到 Map 作业的 worker 读取数据，并将产生的结果缓存在内存中。（4）缓存的结果被定期地写入磁盘。（5）Reduce worker 根据从 Master 发来的信息，从本地磁盘中读取由 Map worker 产生的结果，并对数据排序后进行相应的逻辑处理。（6）Reduce worker 产生的输出会添加到这个分区的最终输出文件中。（7）当所有的 Map 作业和 Reduce 作业完成后，Master 会唤醒用户进程，用户进程中对 MapReduce 的调用返回到用户代码。

在整个 MapReduce 任务的执行过程中，作用调度是通过 Hadoop 自动完成的。

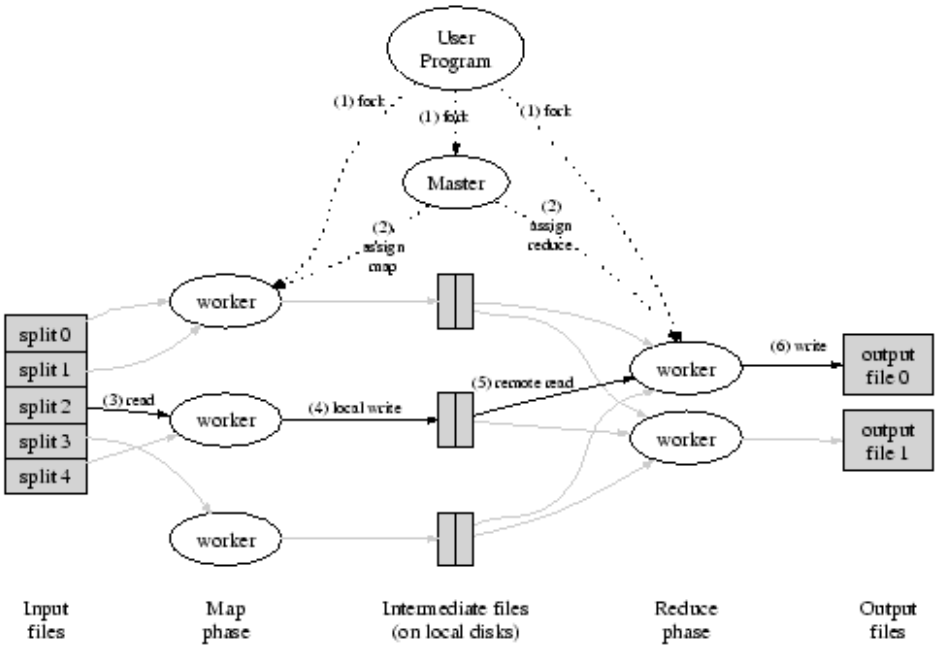



图 2-2：Map/Reduce 任务执行过程[5]

## 2.3 网络爬虫技术

现在，搜索引擎在人们的生活中扮演着越来越重要的角色。网络爬虫是搜索引擎最基础的部分，许多搜索引擎都分别构建了自己的网络爬虫，如百度的Baidu Spider，谷歌

[5] 本图来源于论文《MapReduce: Simplified Data Processing on Large Clusters》，参考文献[26]

的GoogleBot等。

下面对两个开源的网络爬虫进行对比介绍：

（1） Heritrix

从图2-3中可以清楚地看到，Heritrix主要由三大部件组成，分别是：范围控制器、调度器和处理器链。

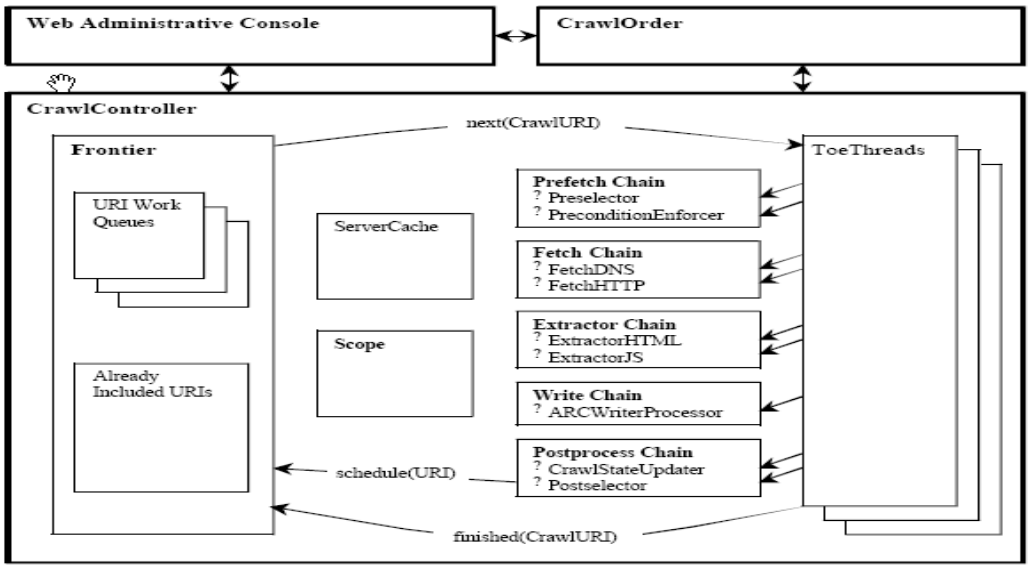



图2-3：Heritrix体系架构图[6]

Heritrix的工作流程主要分为5个步骤：（1）范围控制器根据规则判断是否让新的URI进入预定队列；（2）调度器从预定队列中选择一个要处理的URI；（3）判断URI是否符合抓取条件，是则进入下一步，否则回第一步；（4）处理器链中的处理器完成网页数据下载、数据分析和归档、抽取新URI等任务；（5）标记已经处理过的URI。

Heritrix十分容易扩展，图中的CrawlController中所有部件都开放了接口，可以按需扩展，并通过配置选用相应的部件。

（2） Nutch

Nutch是一个开源的搜索引擎框架，从它的体系结构图中可以看出，它由网络爬虫和搜索两部分组成。

Nutch爬虫的工作流程如图2-4所示，主要分为6个步骤：1）创建一个新的WebDB，并写入种子URI；2）根据WebDB生成抓取链，并写入相应的segment；3）根据抓取链中

[6] 本图来源于网络，http://imgsrc.baidu.com/baike/pic/item/4e0b3ea4414216b49052ee81.jpg

的URI下载相应的网页数据；4）对下载的网页数据进行解析，得到数据及新的URI；5）根据从网页中提取的新URI更新WebDB；6）循环执行2-5步直至达到预先设定的抓取深度或WebDB已不再生成新的抓取链。

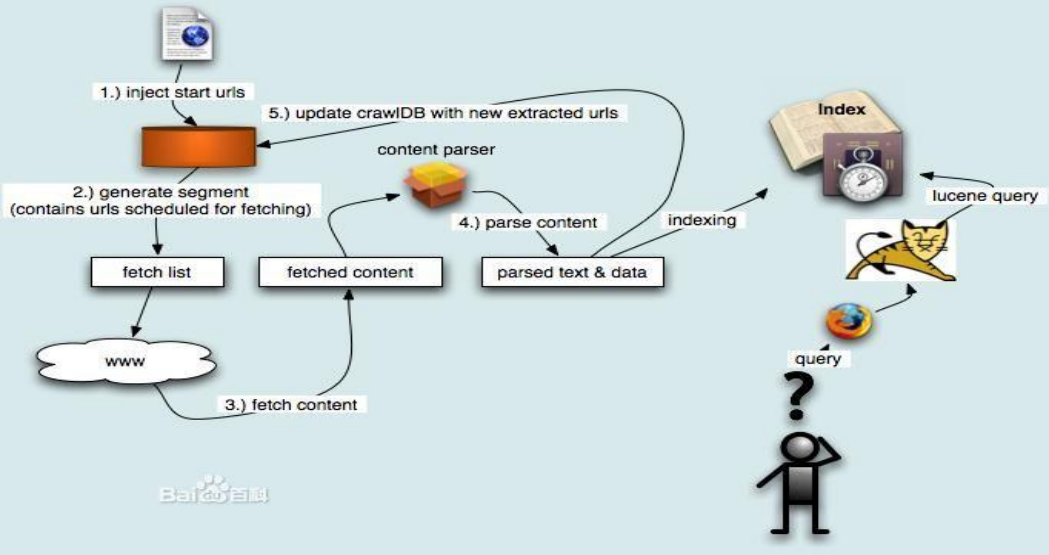



图2-4：Nutch工作流程图[7]

Nutch 的扩展是通过插件机制来实现的，按照规则实现相应的扩展点，并完成相应的配置，即可在 Nutch 中实现对扩展功能的使用。每个 Nutch 进程只能执行一个分布式抓取任务。

（3） 对比分析

本论文对 Heritrix 和 Nutch 的抓取速度做了简单的对比实验：

**实验环境：**

分布式集群：10 台机器

操作系统：CentOS6.5

内存：20GB/台

硬盘：4T/台

CPU：主频 2.33GHz，四核，2 个/台

网络带宽：32Mbit/s

[7] 本图来源于网络，
http://f.hiphotos.baidu.com/baike/c%3DbaikeA1%2C10%2C95/sign=08ed0f7f1038534398cfd070fa78d53f/3812b31bb051f819bbdfff59dab44aed2f738bd4b31cc717.jpg

**实验结果：**

表 2-1：Heritrix 最大线程数 50，Nutch 使用 1 个 map 任务进行抓取

| 文档数（个）<br>抓取时间（秒） | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| Heritrix | 14 | 35 | 56 | 77 | 98 |
| Nutch | 260 | 263 | 265 | 270 | 273 |

表 2-2：Heritrix 最大线程数 50，Nutch 使用 5 个 map 任务进行抓取

| 文档数（个）<br>抓取时间（秒） | 50 | 200 | 500 | 800 | 1000 |
|---|---|---|---|---|---|
| Heritrix | 98 | 379 | 1637 | 2683 | 3365 |
| Nutch | 225 | 336 | 455 | 496 | 640 |

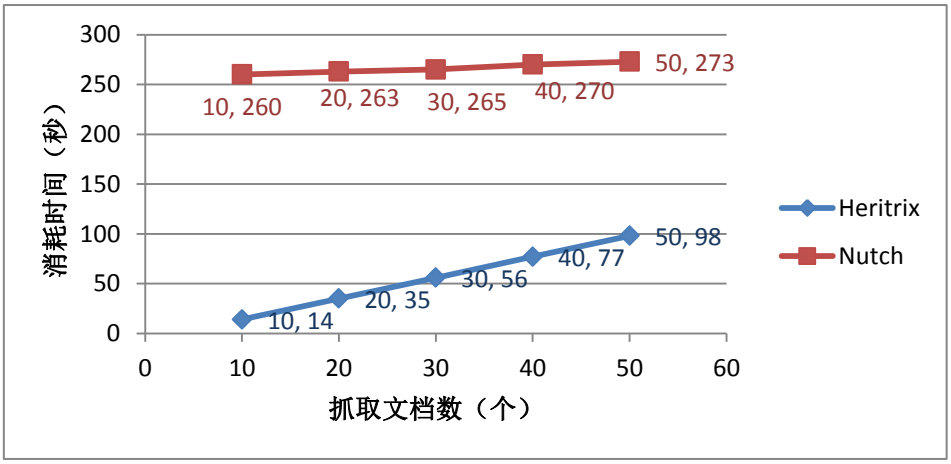



图 2-5：Heritrix 最大线程数 50，Nutch 使用 1 个 map 任务时的对比图

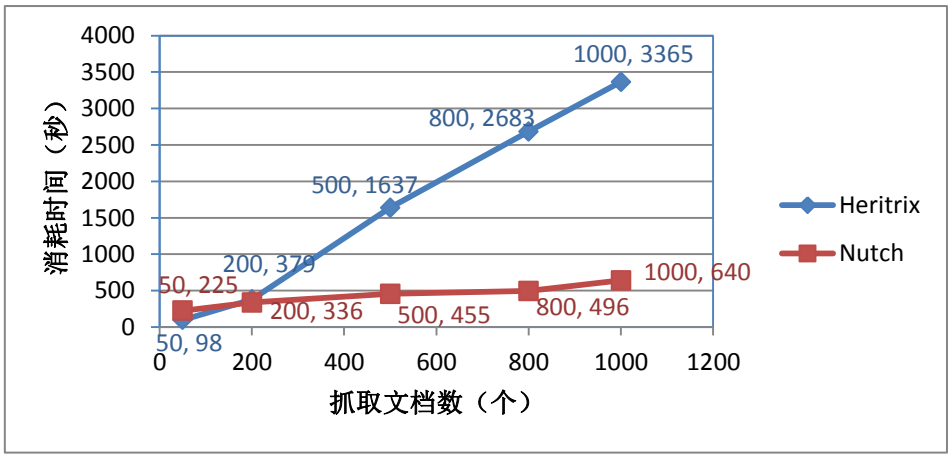



图 2-6：Heritrix 最大线程数 50，Nutch 使用 5 个 map 任务时的对比图

可以看到，随着抓取文档数的增加，Nutch 抓取一个网页的平均时间减少；并且 Nutch 使用 5 个 map 任务进行抓取的速度比 Heritrix 使用 50 个线程的抓取速度要快。综合上述实验结果，表 2-3 给出了 Heritrix 和 Nutch 各方面的简单对比。

表 2-3：Heritrix 和 Nutch 的简单对比

| | | 开源爬虫 | |
|---|---|---|---|
| | | Heritrix | Nutch |
| 运行平台 | | Windows/Linux | Windows/Linux |
| 开发语言 | | Java | Java |
| 各方面对比 | 功能 | 纯粹的网络爬虫，仅提供网站资源的镜像下载功能 | 完整的搜索引擎框架，包括爬虫及搜索两个部分 |
| | 效率 | 实验证明，单机运行时效率稍高于 Nutch | 分布式运行时速度远快于 Heritrix |
| | 结构 | 结构清晰，且模块间耦合度较低 | 包含爬虫和查询两个部分，爬虫部分耦合度比 Heritrix 高 |
| | 分布式 | 不支持 | 使用 Map Reduce 编程方式，很好地支持了分布式处理 |
| | 配置 | 提供了 Web UI，每个任务拥有自己独立的配置文件，可动态修改，支持多任务运行 | 仅提供 CLI，配置文件需在程序运行前配置，单控制界面不支持多任务运行 |
| | 扩展性 | Heritrix 的三大部件都开放了自己的接口，可以很容易地进行扩展 | Nutch 的扩展采用插件机制，可根据实际需要选择相应的扩展点进行功能扩展 |
| | 操作性 | Heritrix 可以在任务执行过程中控制其暂停、开始、结束。 | Nutch 对任务执行缺少控制方式，一旦运行只能等待任务运行完成或手动中止线程 |

从表 2-3 的对比可以看出，Heritrix 和 Nutch 的优缺点都较为明显。Heritrix 的可配置性和可控制性要优于 Nutch；而 Nutch 在分布式集群下可以充分发挥其分布式抓取和处理的速度优势。根据 iSearch 系统的需求，要求快速并且大量地抓取某个或某类网站的数据，无需频繁地对配置进行更改，对抓取任务执行暂停、重启的需求并不明显；出于这些方面的考虑，Nutch 更符合 iSearch 系统的抓取要求。

Nutch仅提供了对网页的整页下载，可通过Nutch的插件机制来进行功能扩展，以满足提取特定信息的需求。这将在下一节中进行详细介绍。

## 2.4 网页信息抽取技术

在前文中已经提到，现有的网页信息提取技术无法满足 iSearch 系统的要求。本文针对 iSearch 系统的信息提取需求，结合 XPATH 技术提出了相应的解决方案。

### 2.4.1 XPATH

XPath[29]即 XML Path Language，它是一门在 XML 文档中查找信息的语言，用于在 XML 文档中通过元素和属性进行导航。XPth 是 W3C XSLT 标准和主要元素，包含一个标准函数库。下面对 XPath 的路径表达式、运算符两个方面进行简单介绍。

（1） 路径表达式

路径表达式与常规文件系统的文件路径十分相似，XPath 通过路径表达式来选取 XML 文档中的节点或节点集。它是从一个 XML 节点到另一个节点的步骤顺序描述，例如“/root”表示选取根元素 root，“book”表示选取 book 节点的所有子节点。表 2-4 列出了最常用的路径表达式。

表 2-4：XPath 常用路径表达式[8]

| 表达式 | 描述 |
|---|---|
| nodename | 选取 nodename 节点的所有子节点 |
| / | 从根节点开始选取，若路径起始于“/”，则表示此路径为绝对路径 |
| // | 选择匹配的节点，不考虑它们的位置。如//book，表示选择文档中所有 book 元素；bookstore//book 表示选择属于 bookstore 元素后代的所有 book 元素。 |
| . | 选取当前节点 |
| .. | 选取当前节点的父节点 |
| @ | 选取属性，如//@lang，表示选取名为 lang 的所有属性 |

（2） 运算符

XPath 表达式可以进行加减乘除等算术运算，还支持与、或等逻辑运算。例如表达式“a>9 and a<8”，返回值一定为 false。

XPath 还提供了超过 100 个内建函数，包括字符串、数值、布尔值、日期时间等数据类型的处理。

[8] 参照 http://www.w3school.com.cn/xpath/xpath_syntax.asp

### 2.4.2 基于 XPATH 的模板信息提取方法

下面通过一个简单的示例来说明本文所提出的信息提取方法的模板文件编写方法和实际工作流程及效果[9]，例中所用的网页链接为“http://movie.mtime.com/156424/”，下文以链接 A 代替：

（1）从链接 A 的网页中找到感兴趣的信息，如图 2-7 所示。三个红色方框标记的是需要提取的内容，其中①和②为需要提取的文本内容，而③为需要提取的链接，下文以链接 B 代替。

（2）根据（1）找出的内容，通过网页源代码来分别找出它们所对应的 XPATH 路径表达式。在本系统实现过程中，笔者使用 FireBug 来帮助编写 XPATH 路径表达式[10]。下面以①为例说明 XPATH 路径表达式的编写方法。


图 2-7：链接 A 对应网页

[9] 本文提出的网页信息提取方法要求模板编写人员具有一定的 HTML 和 XPATH 背景知识。

[10] FireBug 是 FireFox 浏览器的一个插件，可以浏览网页的源代码并且进行调试。


图 2-8：链接 A 页面元素及对应源代码

从图 2-8 中可以看到，①标出的字符串“通灵男孩诺曼 ParaNorman”分布在两个 span 节点中，而这两个 span 节点都是同一个 h1 节点的子节点，因此可简单地通过路径表达式//h1/span 来获取电影名的两个 span 节点。但这样过于简单的路径表达式可能会带来这样的问题：网页中满足条件的节点可能多于你希望提取的节点。为了避免提取到多余的数据节点，笔者建议利用节点的属性来帮助 XPATH 准确选取真正感兴趣的节点。比如在本例中，①对应的 span 节点的父节点 h1 的 class 属性值在网页中是唯一的，通过该属性值可以准确而且唯一地定位到①的真正父节点，修改后的 XPATH 路径表达式为“//h1[@class='movie_film_nav normal pl9 pr15']/span”。

根据上述方法，得到②标记的字符串“克里斯·巴特勒　山姆·菲尔”的 XPATH 路径表达式为“//ul[@class='lh20']/li/strong[contains(text(),'导演')]/..”，而③标记的链接 B 的路径表达式为“//div[@pan='M08_Movie_Overview_Poster']/p/a”。

需要提到的是，由于动态网页的结构几乎不可能完全相同，因此在编写路径表达式时应尽量避免使用绝对路径，否则会因为网页节点的增删造成 XPATH 路径表达式获取节点时出错。

（3）根据（2）得到的 XPATH 路径表达式，编写模板文件，如图 2-9。

```xml
<?xml version="1.0" encoding="UTF-8"?>
<websites>

    <!-- mtime 时光网影片详细页面 -->
    <website domain="com.mtime.movie">
        <url-pattern>http://movie.mtime.com/*/</url-pattern>
        <data>
            <!-- occur value must in {"mandatory", "should", "optional"}, default is "optional" -->
            <field name="name" multi-value="false" occur="should" description="电影名">
                <template>
                    <expression>//h1[@class='movie_film_nav normal pl9 pr15']/span</expression>
                    <range end="加关注" />
                </template>
            </field>

            <field name="director" multi-value="false" occur="should" description="导演">
                <template>
                    <expression>//ul[@class='lh20']/li/strong[contains(text(),'导演')]/..</expression>
                    <range start="：" end="更多" />
                </template>
            </field>
        </data>

        <outlink>
            <entry occur="mandatory" description="imgs link">
                <template>
                    <expression>//div[@pan='M08_Movie_Overview_Poster']/p/a</expression>
                </template>
            </entry>
        </outlink>
    </website>
```

图 2-9：基于 XPATH 的模板信息提取方法所使用的模板文件内容示例

图 2-9 中属性值 name=”name”的 field 节点对应例子中的①的信息提取规则，属性值 name=”director”的 field 节点对应例中②的信息提取规则，而 outlink 节点中的 entry 节点对应③的链接提取规则。下面对模板文件的构成进行介绍：

1）expression 节点即是 XPATH 路径表达式在模板文件中的表现形式。

2）range 节点的“start”和“end”属性分别指明了通过 expression 中的 XPATH 路径表达式选取的内容中真正需要提取的文本的开始字符及结束字符。例如，通过 XPATH 路径表达式选取出来的内容是“电影名：阿凡达（2009）”，而实际要提取的内容是“阿凡达”，就可以通过设置 start=”：”，end=”（”来通知程序进行进一步处理，只取“：”后“（”前的字符串作为结果。start 属性和 end 属性无须成对出现，可根据实际需要进行选择性设置。

3）一个 template 节点代表一个提取规则；template 节点中必须包含 expression 节点，即 template 节点中必须对信息提取规则进行定义。

4）field 节点的是图 2-7 中①②所标记的文本提取规则在模板文件中的表现形式，每个要提取的文本内容的规则都对应模板文件中的一个 field 节点。Field 节点有四个属性：name 属性指明提取结果对应的字段名；multi-value 属性指明提取结果是否允许多值，若为“true”则提取结果为数组对象，为“false”则只取结果

数组的第一个值；occur 属性指明该节点的必要程度，“mandatory”代表该节点的提取结果不能为空，否则认为整个网页不符合提取规则，不再对其他节点进行处理、“should”代表该节点的提取结果应该不为空，当结果为空时程序记录警告信息，但不影响其他节点的提取、“optional”代表该节点的提取结果可有可无，不会对提取过程造成影响；desciption 为辅助属性，可用于指明该节点提取的内容类型。每个 field 节点包含一个或多个 template 节点（由于网页结构的变化，可能出现视觉上处于同一位置的节点的 XPATH 路径表达式不同），template 节点决定了 field 节点的文本提取结果。

5）data 节点表示的是一个网页中所有提取的文本内容的集合，由一个或多个 field 节点组成。在本例中 data 包含了两个 field 节点，分别对应图 2-7 中①、②标记内容的提取规则。

6）entry 节点与 field 节点类似，entry 节点对应的是图 2-7 中③所标记的链接 B 的提取规则，通过它的子节点 template 来决定链接提取的结果。

7）outlink 节点则是与 data 节点类似，outlink 节点是一个网页中所有要抽取的链接的集合，由一个或多个 entry 节点组成。

8）url-pattern 结点指明了使用该模板的 URL 匹配规则，“*”号代表通配“/”号以外的所有字符。如本例中的“http://movie.mtime.com/*/”，作用与正则表达式“http://movie.mtime.com/[^/]+/”一致。

9）url-pattern、data 和 outlink 三大结点共同组成了模板，即文件中的 website 结点。只有满足 url-pattern 匹配规则的网页才会使用 data 和 outlink 结点中定义的提取规则进行信息抽取。

本文在前面已经提到，在 iSearch 系统抓取数据的过程中，同一事物的相关信息可能会分布在多个页面上。下面对上例继续进行分析。

在上例中，链接 A 与③标记的链接 B，实际上都是指向包含了同一部电影相关信息的网页，但是在数据抓取过程中，链接 A 和链接 B 会被独立处理。为了支持数据抓取后的数据分析和合并，在页面的数据提取结果中需要包含关联标记。

图 2-10 是链接 B 对应的页面及其源代码。从图中可以清楚地看到，在链接 B 对应的页面中包含了链接 A 字符串，只需将此字符串作为 B 页面信息提取结果的一个字段

即可很方便地将链接 A、B 对应页面的提取结果关联到一起。其他网页的关联方法以此类推。


图 2-10：链接 B 对应页面及源代码

根据上文的描述可以看出，对于不同的网页，由于其结构及包含信息的不同，页面间的关联标记也各不相同；因此，在对提取的数据进行后续的分析和合并时，需要针对不同的关联标记分别进行处理。这是本论文今后需要解决的一大问题。

需要说明的是，一个模板文件实际上包含的是一个模板链。一般情况下，可以将互相关联的网页的提取模板放在同一个模板文件中，这样更便于模板的管理及后续的数据关联分析。

本论文提出的基于 XPATH 的模板信息提取方法根据定制的模板可以同时对多种结构的页面进行分析和信息提取，更符合网络爬虫抓取数据时需要处理多种类型页面的实际需求。但由于无法对所有网页进行分析，基于模板的信息提取方法都无法保证信息提取结果是完全正确的。该方法的具体处理流程如图 2-11 所示。

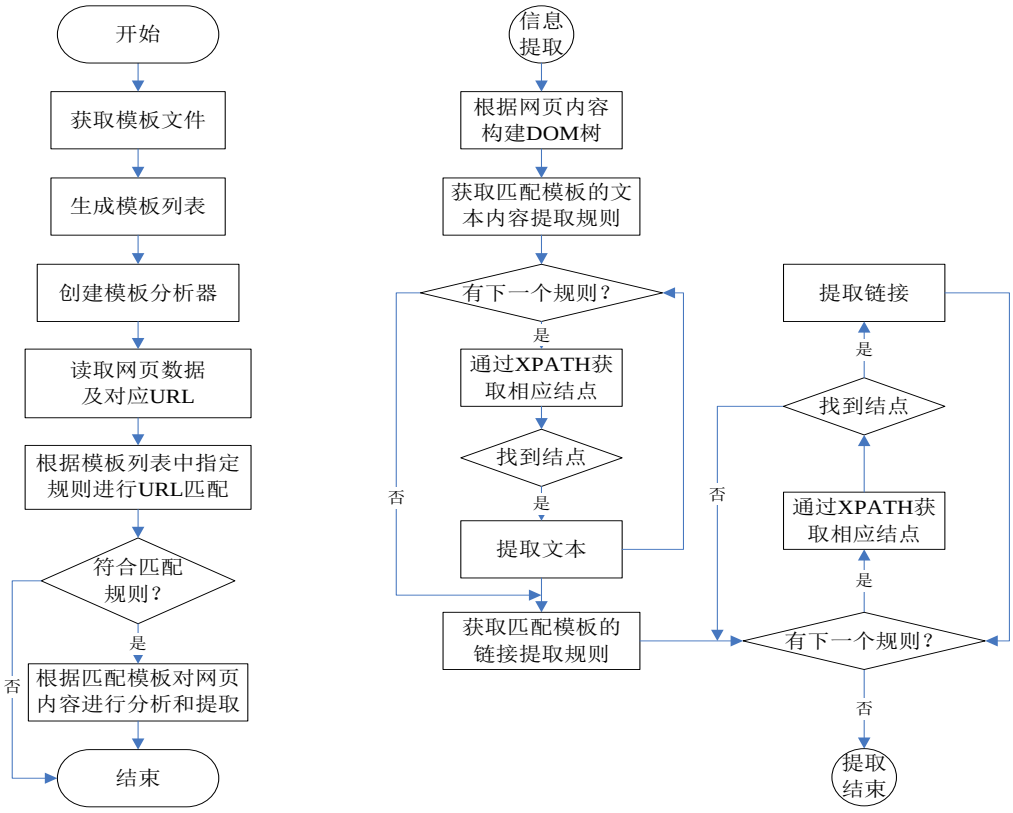



图 2-11：(a)基于 XPATH 的模板信息提取方法处理流程。(b)为(a)中根据 XPATH 路径表达式提取信息的具体流程。

## 2.5 增量索引方案

索引技术是搜索引擎的核心技术之一。对于 iSimilar 图像检索平台来说，构建索引同样是必不可少的步骤。笔者发现，在向 iSimilar 的图片仓库新增一张图片之后，iSimilar 的索引模块会对图片仓库中存储的所有图片重新构建索引，随着图片数量的增多，构建索引所耗费的计算量及时间也随之增大。特别是当频繁地执行插入操作时，甚至会对平台的性能造成影响。而 iSearch 系统提供的广告上传功能需要 iSimilar 平台支持较为频繁的插入操作，当前的 iSimilar 平台不能很好地支持这一需求。

为了解决上述问题，笔者考虑使用 Lucene[30]为 iSimilar 平台提供增量索引功能。LIRe[31]项目也是 Lucene 在图像检索上的一个实际应用。

Lucene 是 Apache 的一个子项目，它是一个开源的全文检索引擎工具包，提供了完整的查询引擎和索引引擎。使用 Lucene 可以很方便地目标系统中实现全文检索功能。

下面对 Lucene 中几个比较重要的类进行简单介绍：

（1） IndexWriter

索引写入器，负责创建及维护索引，可以对索引进行增删改的操作。

（2） Analyzer

词法分析器，指明索引的内容按什么样的方式来建立。

（3） Directory

索引目录的抽象，指明索引存储的目录。

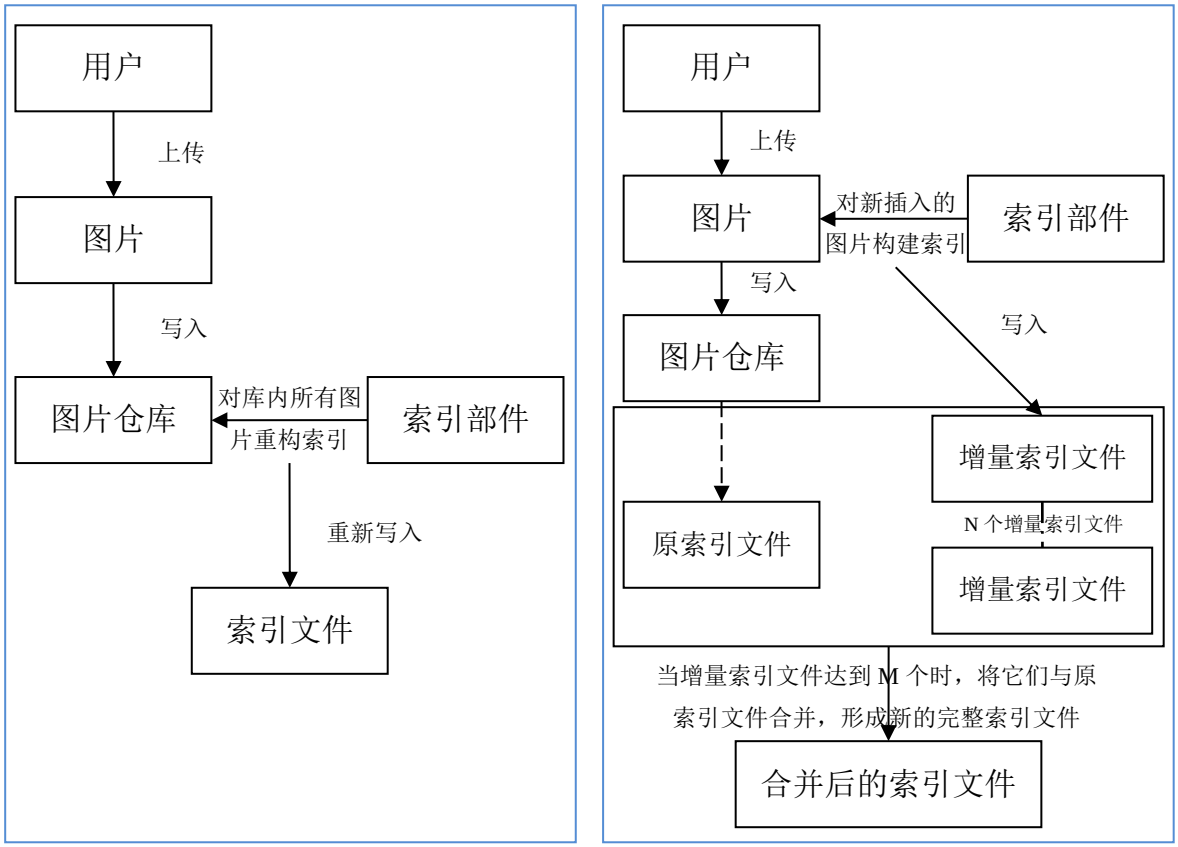
（4） Document

要索引的文档对象，索引中的每一条信息用一个 Document 来表示

（5） Field

Document 中的字段属性，在创建时可指定是否存储该字段值及字段的索引方式。

图 2-12(b)展示了本文提出的增量索引方法工作流程，与 iSimilar 索引方法不同的是，当新的图片插入图片仓库之后，索引模块直接对新插入的图片构建索引，并写入新的索引文件；在索引文件数量达到 M（M 可根据实际需求配置）时，对 M 个索引文件与原索引文件进行合并。




(a) (b)

图 2-12：(a)为 iSimilar 原索引过程；(b)为本论文索引方法过程

## 2.6 SSH 框架

SSH[32]是一种集成了 struts、spring 和 hibernate 的 Web 开源应用框架。使用 SSH 框架搭建 Web 应用程序可以很容易地实现控制器（Controller）、视图（View）和模型（Model）的分离。下面以 iSearch 系统的检索功能为例对 SSH 框架的工作流程进行说明：在表示层中，从 JSP 页面向后台提交搜索请求，Struts 根据配置文件将请求分派给相应的 Action 部件进行处理；在业务逻辑层中，Spring 组件负责向 Action 部件提供业务模型组件及对应的数据处理（DAO）组件以协助完成业务逻辑；在数据持久层中，根据 Hibernate 的对象和数据库的映射，完成 DAO 组件的请求。

## 2.7 AJAX 技术

Ajax[33]即“Asynchronous JavaScript and XML”，这一概念由信息架构师 Jesse James Garrett 在 2005 年提出。但这项技术在 1998 年就已经开始应用。Ajax 并不是一门新的编程语言，而是一种用于创建交互性更强的 Web 应用，改善用户体验的技术。

图 2-13 和图 2-14 分别展示了传统 Web 请求交互模式及 Ajax 请求交互模式。

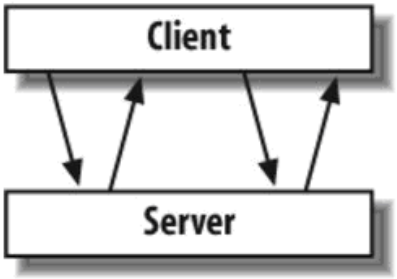



图 2-13：传统 Web 请求模式

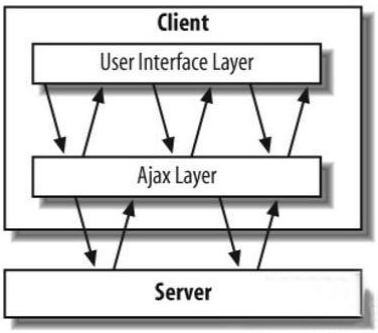



图 2-14：Ajax 请求模式

在传统的 C/S 交互模式中，用户向服务器提交请求之后，等待服务器接受请求并将处理后的结果返回。在此过程中，用户不能再对当前页面进行操作。而当服务器向用户返回处理结果后，无论处理过程成功与否，浏览器都会刷新当前页面。

而使用 Ajax 技术的交互模式比传统模式多了一个 Ajax 层，用户的请求并不直接提交给服务器进行处理，而是通过 Ajax 层将请求内容包装成 Ajax 请求对象再发送到服务器进行处理。而服务器处理请求后也不直接将结果反馈给用户，而是发送到 Ajax 层，

由 Ajax 层根据处理结果来采取相应的措施。在此过程中，浏览器不会对当前页面进行刷新，用户可以继续执行其他操作。

Scott Raymond 在《Ajax on Rails》[33]一书中提到，Ajax 技术有以下几个特性：（1）有些用户请求可以在客户端完成，并不一定要发送到服务器进行处理，如表单验证；（2）Ajax 与服务器之间的只对必需的数据进行交互，响应时间缩短；（3）页面不再直接依赖于服务器的响应，在等待服务器处理结果的同时用户可以对页面进行其他操作；（4）页面与服务器之间的交互不再意味着页面必须进行全部刷新。

互联网的快速发展，网民数量的增多，用户对 Web 应用程序体验的要求越来越高，这为 Ajax 提供了良好的发展空间。正确使用 Ajax 技术不仅可以减少用户的等待时间，而且通过由客户端来完成部分用户请求可以一定程度上减轻服务器的负担。iSearch 系统也采用了 Ajax 技术来改善用户使用体验。

# 第三章　iSearch 系统需求分析

第二章对 iSearch 系统中所涉及的关键技术及概念进行了简单介绍。从本章开始，将对 iSearch 系统的分析和设计展开说明。

iSearch 是建立在 iSimilar 图像检索平台的核心技术和架构之上的广告信息增强系统，它的主要目的是利用可视化图像检索技术，增强海报广告的交互性，以起到加强广告宣传效果的作用。从功能上进行划分，iSearch 系统主要由四个模块组成：Web 端可视化检索模块、广告注册模块、手机端可视化检索模块、数据操作模块。在这四个模块中，数据操作模块是系统最基础的部分，可视化检索功能是系统提供的最重要的功能；系统的分析与设计将重点围绕这两点展开。

在本章中，3.1 小节对 iSearch 系统的整体设计思路简单地进行了阐述；3.2 小节分模块对系统的主要用例进行讨论，重点分析数据操作模块和 Web 端可视化检索模块；在 3.3 小节中对系统领域模型的设计和建模进行讨论；在 3.4 小节中对系统的非功能性需求进行了简单分析。

## 3.1　iSearch 系统概述

移动互联网图像检索技术的发展带动了移动可视化图像检索的发展。在此条件下，iSimilar 平台提出了 iSearch 项目，该项目的主要目的是利用可视化图像检索技术来提高海报广告交互能力、增强广告效果。本论文尝试为 iSearch 项目提出解决方案。该方案将结合网络爬虫、网页信息提取、手机应用程序开发、Web 应用程序开发等多项技术，构建一个端到端的广告信息增强系统。

iSearch 广告信息增强系统的整体设计思路如下：（1）分类快速搜集大量图片及相关描述信息，建立基础数据仓库；（2）构建广告注册模块，提供用户注册、登录、退出、信息修改、上传广告管理，人工上传图片并添加相关扩展信息等功能，用户必须通过注册并登录才能使用本模块的功能；（3）构建可视化图像检索模块，用户可以方便地从 Web 浏览器或手机客户端获取可视化检索服务。快速搜集数据的功能通过网络爬虫技术来实现；本论文选取 Nutch 分布式爬虫做为基础，加入了本论文提出基于 XPATH 的模

板信息提取方法，构建了一个可定制的快速提取指定信息的分布式网络爬虫。账户管理模块避免了用户随意更改其他用户上传的广告数据，一定程度上加强了数据安全性。

## 3.2 iSearch 系统主要用例分析

根据上述整体设计思路，系统可分为四大模块：（1）数据操作模块，包括网络爬虫数据抓取、数据分析处理、数据统计、模板管理等功能；（2）Web 端可视化检索模块，提供通过浏览器获取 iSearch 可视化检索服务的功能；（3）手机客户端，提供通过手机获取 iSearch 移动可视化检索服务的功能；（4）广告注册模块，管理用户注册、登录及退出系统，提供人工上传图片并标注相关扩展信息及对本账户上传图片进行管理的功能。

iSearch 系统主要有三类用户，一是使用可视化检索服务的用户，本论文中将其称为搜索用户；二是使用广告上传及账户管理功能的用户，称为广告用户；还有一类是为系统提供数据来源及系统维护的用户，称为系统用户。下面将分模块对主要用例进行分析。

### 3.2.1 数据操作模块主要用例

数据操作模块包括了网络爬虫数据抓取、数据分析处理、数据统计、模板管理等子模块。在本小节中，用户都是指系统用户。下面对四个子模块分别进行分析。

（1） 网络爬虫子模块

在使用该子模块的时候，用户可能执行的操作有：1、对当前爬虫的配置进行修改。2、新建包含种子 URI[11]的文件。3、执行一个新的抓取任务。4、停止一个正在运行的抓取任务。5、添加一个新的扩展功能。

对以上操作分析得到网络爬虫子模块的基本用例，包括开始抓取任务、停止抓取任务、修改爬虫配置、新建种子文件、添加扩展功能。下面对以上四个用例进行分析。其中停止抓取任务用例使用摘要形式进行描述，新建种子文件和添加扩展功能用例使用非正式形式进行描述，开始抓取任务和修改爬虫配置用例使用详述形式进行描述（见表 3-1 和表 3-2）。网络爬虫子模块的用例图如图 3-1 所示。

[11] 种子 URI 指初始状态下，由用户提供的网络爬虫需要抓取的网页地址。

**摘要用例：停止抓取任务**

基本事件流：用户从正在执行的抓取任务中找到要停止的任务，记录任务的 jobID。打开系统命令行终端，执行命令 1“hadoop job –kill jobID”[12]。Hadoop 系统接收请求并通过 jobID 找到相应的活动任务，停止其进程。如果用户请求停止的任务不存在或已执行完毕，Hadoop 不向用户返回提示，默认请求处理成功。

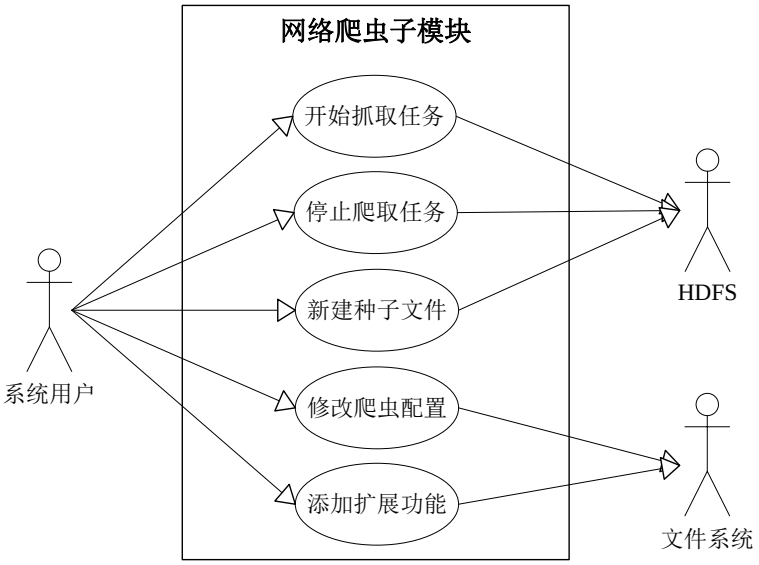



图 3-1：网络爬虫子模块用例图

**非正式用例：新建种子文件**

基本事件流：用户搜集需要提交给爬虫进行抓取的网页地址，以每行一个网页地址的格式写入文本文件中[13]并保存，文件路径为 path。打开系统命令行终端，执行命令 2“hadoop fs –put path hdfsPath”，path 为种子文件路径，hdfsPath 为种子文件在 HDFS 文件系统中的保存路径。若 HDFS 中不存在路径为 hdfsPath 的文件，则新建该文件，并将种子文件的内容复制写入。

备选路径：若 HDFS 中已经存在路径为 hdfsPath 的文件，Hadoop 向用户返回操作失败提示。用户可以修改 hdfsPath 或删除 HDFS 上路径为 hdfsPath 的文件，重新执行命令 2。

**非正式用例：添加扩展功能**

基本事件流：用户将编写好的扩展功能插件放入网络爬虫的插件文件夹下，根据需要对网络爬虫相应的配置文件进行修改并保存。用户通过命令行终端对网络爬虫进行重订报编译及打包。命令行终端输出提示编译打包成功。

备选路径：命令行终端输出编译打包失败提示。用户根据错误提示进行修改，并重新编译打包网络爬虫。

[12]命令中的 jobID 即前文记录的任务的 jobID

[13]以“#”号开头的行是注释行，爬虫不对其进行处理

下面对开始抓取任务、修改爬虫配置两个用例进行详述。

表 3-1：开始抓取任务用例详述

| | |
|---|---|
| 用例名 | 开始抓取任务 |
| 用例概述 | 用户执行一个抓取任务 |
| 使用频率 | 中，基础功能，为系统提供数据准备 |
| 主要参与者 | 用户 |
| 前置条件 | 用户打开计算机 |
| 基本事件流 | 1、 用户打开系统的命令行终端。<br>2、 用户通过“cd”命令进入网络爬虫执行文件所在目录。<br>3、 用户在命令行终端中输入启动网络爬虫抓取的命令，并执行，命令中需指定网络爬虫需要读入的种子文件的路径。<br>4、 任务正常启动，爬虫自动抓取网页数据并对下载的数据根据配置文件中指定的模板进行分析和信息提取，并将提取后的数据写入 HDFS。<br>5、 命令行终端输出爬虫抓取任务执行状态，Hadoop 系统更新任务列表。<br>6、 完成 |
| 备选路径 | 3a、请求进入的路径错误。<br>1、 命令行终端输出错误提示。<br>2、返回上一步。<br>4a、调用的命令格式不正确。<br>1、命令行终端输出相应的错误信息，可根据提示查看爬虫调用命令说明。<br>2、返回上一步。<br>4b、种子文件路径不存在。<br>1、命令行终端输出提示种子文件路径不存在。<br>2、返回上一步。 |
| 后置条件 | 抓取任务开始执行，Hadoop 系统更新任务列表 |
| 特殊需求 | 用户使用的计算机使用 Linux 操作系统，安装 hadoop1.0.3，并正确配置 |

为了清楚地展示网络爬虫抓取任务执行过程，通过图 3-2 的数据流图进行描述。

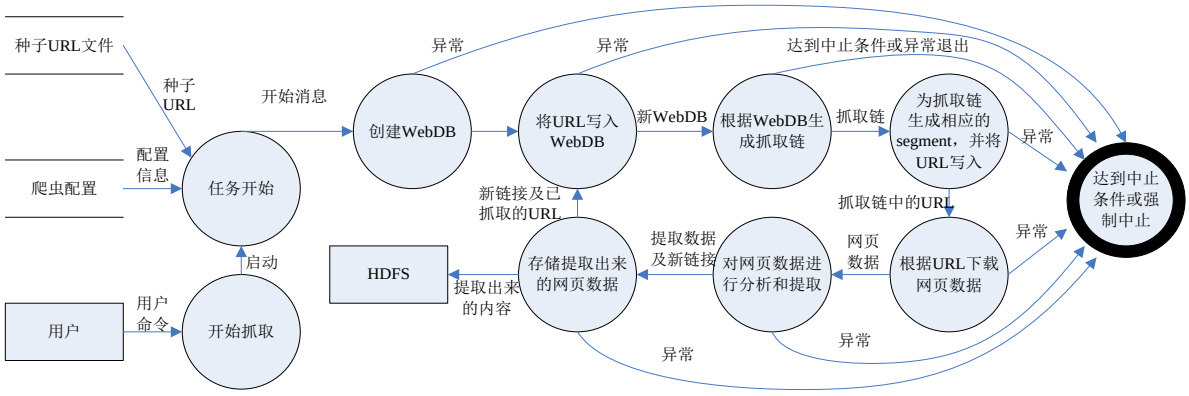



图 3-2：网络爬虫抓取过程数据流图

表 3-2：修改爬虫配置用例详述

| | |
|---|---|
| 用例名 | 修改爬虫配置 |
| 用例概述 | 用户对当前爬虫的配置进行修改 |
| 使用频率 | 中，用户根据需要选择使用 |
| 主要参与者 | 用户 |
| 前置条件 | 要修改配置的爬虫当前没有正在执行任务 |
| 基本事件流 | 1、用户根据实际需要修改爬虫的配置文件并保存。<br>2、用户打开系统的命令行终端。<br>3、用户通过“cd”命令进入网络爬虫的根目录。<br>4、用户在命令行终端执行命令对网络爬虫进行重新编译和打包。<br>5、编译及打包网络爬虫成功，命令行终端输出相应提示。<br>6、用户在命令行终端执行命令“ls -l”查看网络爬虫包含目录及文件操作权限。<br>7、用户对网络爬虫包含的所有目录及文件都有“x”即执行权限。<br>8、完成。 |
| 备选路径 | 5a、编译及打包网络爬虫失败。<br>1、命令行终端输出错误提示。<br>2、用户根据提示检查并进行修改。<br>3、返回上一步。<br>7a、用户缺少对网络爬虫目录或文件的执行权限。<br>1、使用“cd”命令进入网络爬虫的根目录。<br>2、执行命令“chmod –R +x ./”。<br>3、完成。 |
| 后置条件 | 配置修改成功，网络爬虫重新编译及打包成功 |
| 特殊需求 | 用户使用的计算机使用 Linux 操作系统，安装 ant1.8.x |

（2） 数据分析处理子模块

在数据分析处理子模块中，用户可能执行的操作有：执行数据分析和合并、统计图片数量、统计电影数量。本子模块的用例都使用非正式形式进行描述。用例图如图 3-3 所示。

**非正式用例：数据分析与合并**

基本事件流：用户通过命令行终端调用相应的数据分析与合并功能，Hadoop 根据用户请求启动相应的任务，并把处理结果存储到 HDFS 上用户指定的路径。

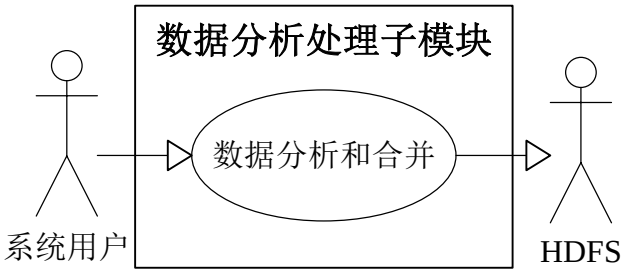



图 3-3：数据分析处理子模块用例图

备选路径：用户调用命令有误，Hadoop 不能成功启动任务，命令行终端输出相应提示。

（3） 数据统计子模块

该子模块主要提供对抓取电影数据的图片数及电影数的统计功能，有统计电影数量和统计图片数量两个基本用例，在本小节中用非正式用例形式描述。用例图如图 3-4 所示。

**非正式用例：统计图片数量**

基本事件流：用户通过命令行终端调用图片数量统计功能，命令中需要指定需要处理的数据所在路径及处理结果的输出路径。Hadoop 接收请求并启动相应的任务，根据用户指定的输入文件路径读取数据，并将处理结果写入用户指定的输出路径。

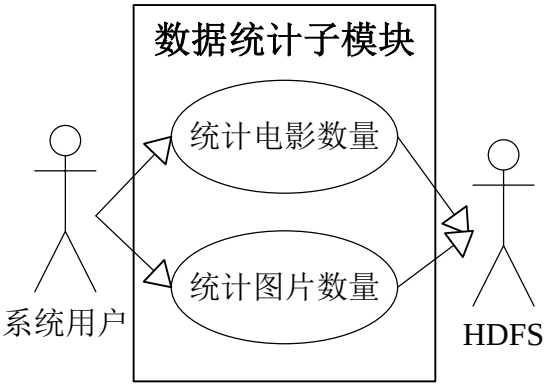



图 3-4：数据统计子模块用例图

备选路径：如果用户指定的输入文件路径不存在，命令行终端输出相应错误提示。如果用户指定的输出路径已存在，命令行终端输出错误提示。

**非正式用例：统计电影数量**

基本事件流：用户通过命令行终端调用电影数量统计功能，命令中需要指定需要处理的数据所在路径及处理结果的输出路径。Hadoop 接收请求并启动相应的任务，根据用户指定的输入文件路径读取数据，并将处理结果写入用户指定的输出路径。

备选路径：如果用户指定的输入文件路径不存在，命令行终端输出相应错误提示。如果用户指定的输出路径已存在，命令行终端输出错误提示。

（4） 模板管理子模块

在模板管理子模块中，用户可能执行的操作有：新建模板文件、修改模板文件、浏览模板文件、删除模板文件、使用网页对模板的提取效果进行测试。分析得到相应的五个基本用例，其中浏览模板文件和删除模板文件用例使用非正式形式描述，新建模板文件、修改模板文件和模板文件测试使用详述形式描述。用例图如图 3-5 所示。

**非正式用例：浏览模板文件**

基本事件流：用户访问模板管理模块，从当前的模板文件列表中选择一项，提交浏览模板文件的请求。系统接收请求并读取相应的模板文件内容，并将内容输出到页面。

备选路径：如果用户请求查看的模板文件不存在，系统提示用户请求浏览的模板文件不存在或已被删除。

**非正式用例：删除模板文件**

基本事件流：用户访问模板管理模块，从当前的模板文件列表中选择一项，提交删除模板文件的请求。系统接收请求并从 HDFS 删除相应的模板文件，最后刷新模板文件列表。

备选路径：如果用户请求删除的模板文件不存在，系统向用户输出提示模板文件不存在或已被删除。

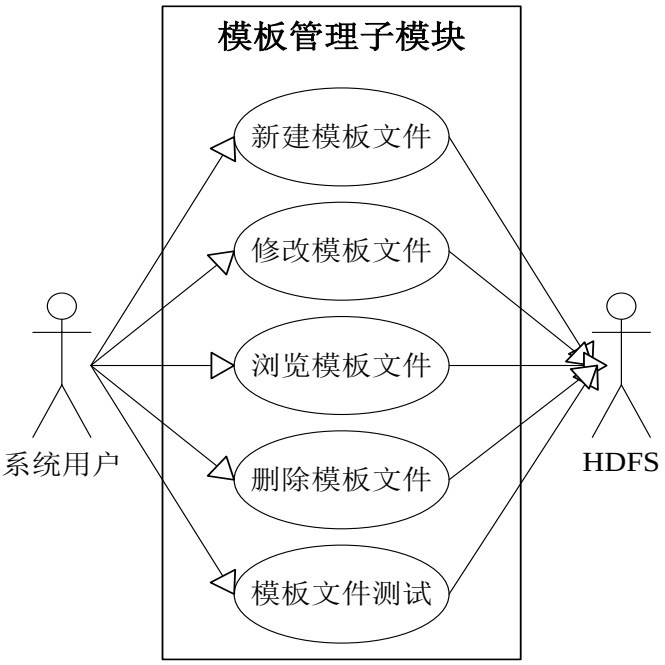



图 3-5：模板管理子模块用例图

表 3-3 至表 3-5 分别对新建模板文件、修改模板文件和模板文件测试三个用例进行了详细描述。

表 3-3：新建模板文件用例详述

| | |
|---|---|
| 用例名 | 新建模板文件 |
| 用例概述 | 用户根据需要新建网页内容提取模板文件 |
| 使用频率 | 中，用户根据需要选择使用 |
| 主要参与者 | 用户 |
| 前置条件 | 用户进入模板管理模块 |
| 基本事件流 | 1、 用户提交新建模板文件请求。<br>2、 系统返回模板填写页面。<br>3、 用户根据页面提示填定相应内容。<br>4、 填写完成后，用户提交确认新建模板文件请求。<br>5、 系统检查用户输入并新建模板文件保存用户输入内容（该步骤具体过程与修改模板文件用例基本事件流步骤 6 至 9 一致，详见表 3-4）。<br>6、 完成。 |
| 备选路径 | 4a、用户提交取消新建模板文件请求。<br>1、系统不进行新建模板操作，返回上一页。<br>2、完成。 |
| 后置条件 | 用户请求成功提交到服务端，服务端进行了正确的处理 |
| 特殊需求 | 网络稳定 |

表 3-4：修改模板文件用例详述

| | |
|---|---|
| 用例名 | 修改模板文件 |
| 用例概述 | 用户根据需要对已有的网页内容提取模板文件进行修改 |
| 使用频率 | 中，用户根据需要选择使用 |
| 主要参与者 | 用户 |
| 前置条件 | 用户进入模板管理模块 |
| 基本事件流 | 1、 用户从当前已有的模板文件列表中选择一项，提交修改模板文件请求。<br>2、 系统根据用户请求读取相应的模板文件。<br>3、 系统将读取的模板文件内容输出到用户页面。<br>4、 用户根据需要对模板进行修改。<br>5、 修改完毕后用户提交确认修改模板文件请求。<br>6、 系统检查用户输入并确认必填项填写完整且正确。<br>7、 系统接收用户请求并检查要写入的模板文件是否已存在。<br>8、 模板文件不存在，新建模板文件。<br>9、 将用户输入内容保存到模板文件。<br>10、完成。 |
| 备选路径 | 3a、请求修改的模板文件不存在。<br>1、系统提示用户请求修改的模板文件不存在。<br>2、系统刷新模板文件列表。<br>3、完成。<br>5a、用户提交取消修改模板文件请求。<br>1、系统不执行保存修改操作。<br>2、返回上一页，完成。<br>6a、必填项没有填写完整或不正确。<br>1、系统向用户返回相应的提示。<br>2、返回第 4 步。<br>8a、模板文件已存在。<br>1、删除已存在的相应模板文件。<br>2、新建模板文件。<br>9a、向模板文件写入内容时发生异常。<br>1、系统向用户返回相应提示。<br>2、刷新模板文件列表。<br>3、完成。 |
| 后置条件 | 用户请求成功提交到服务端，服务端进行了正确的处理 |
| 特殊需求 | 网络稳定 |

表 3-5：模板文件测试用例详述

| | |
|---|---|
| 用例名 | 修改模板文件 |
| 用例概述 | 用户根据需要对已有的网页内容提取模板文件进行修改 |
| 使用频率 | 中，用户根据需要选择使用 |
| 主要参与者 | 用户 |
| 前置条件 | 用户进入模板管理模块 |

| | |
|---|---|
| 基本事件流 | 1、 用户从当前已有的模板文件列表中选择一项，提交测试模板文件请求。<br>2、 系统根据用户请求读取相应的模板文件。<br>3、 系统将读取的模板文件内容输出到用户页面相应位置。<br>4、 用户从本机上传一个用于测试的网页。<br>5、 系统获取用户提交的文件并输出到页面相应位置。<br>6、 用户提交测试请求。<br>7、 系统接收用户请求并将处理结果输出到页面相应位置。<br>8、 完成。 |
| 备选路径 | 2-3a、用户请求测试的模板文件不存在。<br>1、系统提示用户请求测试的模板文件不存在。<br>2、系统刷新模板文件列表。<br>3、返回第 1 步。<br>4a、用户将要用于的网页地址填在系统页面的检索栏，提交查看网页请求。 |
| 后置条件 | 用户请求成功提交到服务端，服务端进行了正确的处理 |
| 特殊需求 | 网络稳定 |

使用活动图对上述用例主要业务流程做进一步描述，如图 3-6。

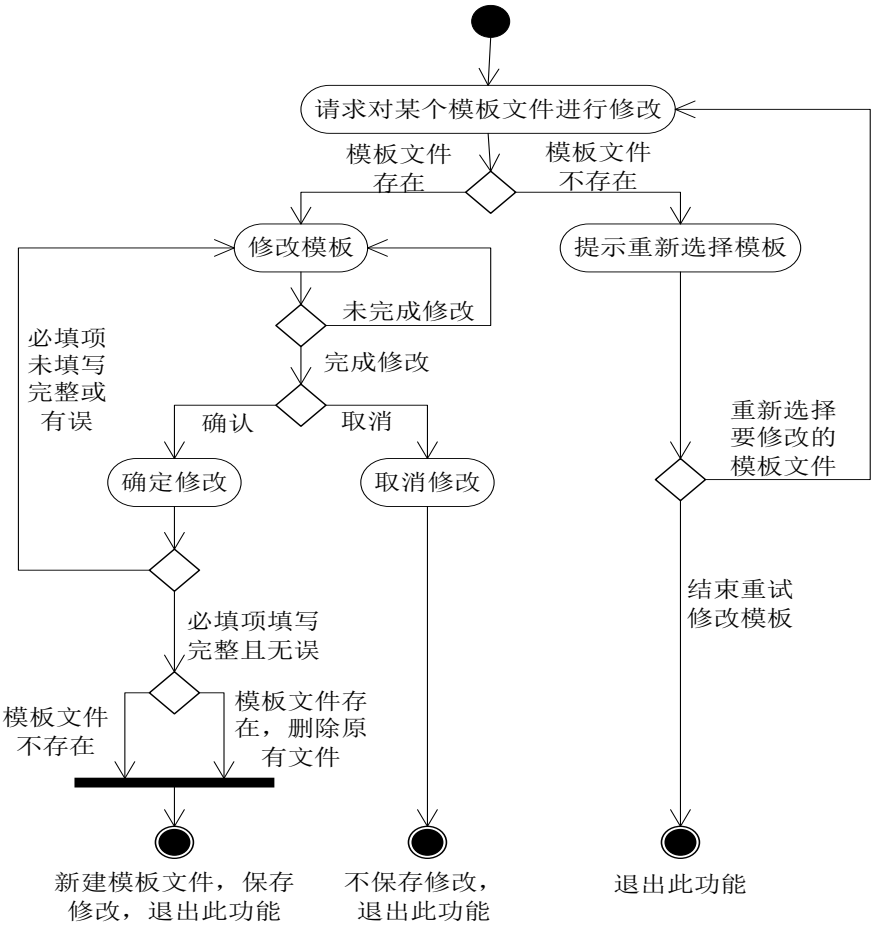



图 3-6：修改模块活动图

### 3.2.2 Web 端可视化检索模块

本模块的用户都是搜索用户。使用本模块时用户可能涉及的操作有：1、上传图片到系统进行检索；2、选择对检索结果的其中一项进行浏览。用例图如图 3-7 所示。

查看结果详情用例使用非正式形式描述，表 3-5 对图片检索用例进行详细描述。

**非正式用例：查看结果详情**

基本事件流：用户从系统返回的检索结果列表中选择一项，请求查看详细信息。系统接收请求并获取相应的信息，输出到用户页面。如果当前频道是广告频道，用户可以通过点击页面中提供的相关链接来获取更多扩展信息。

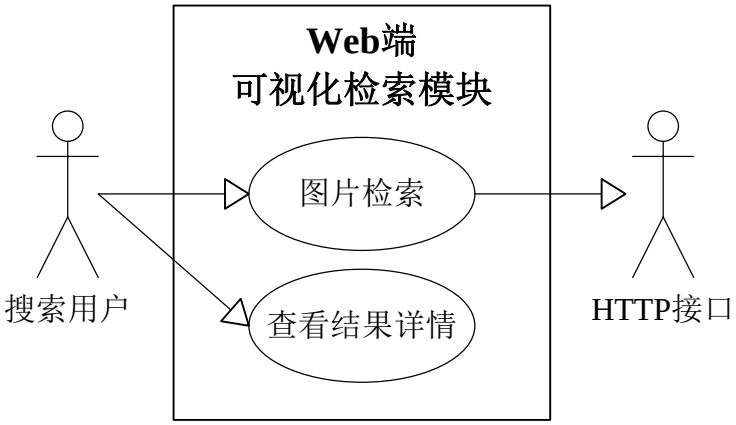



图 3-7：Web 端可视化检索模块用例图

备选路径：如果当前频道是电影频道，用户可以查看电影的影评、微影评、海报图片等信息。

表 3-6：图片检索用例详述

| | |
|---|---|
| 用例名 | 图片检索 |
| 用例概述 | 用户利用系统的可视化检索功能来检索相似图片 |
| 使用频率 | 高 |
| 主要参与者 | 用户 |
| 前置条件 | 用户进入系统的 Web 可视化检索模块 |
| 基本事件流 | 1、 用户从系统提供的可选频道中选择一项。<br>2、 用户从本地计算机中选择要搜索的图片，提交图像检索请求。<br>3、 系统确认用户提交的文件格式正确。<br>4、 系统根据用户请求获取用户请求检索的图片流。<br>5、 系统通过 http 接口从 iSimilar 平台获取检索结果。<br>6、 系统向用户输出用户提交查询的图片及查询结果列表。<br>7、 完成。 |
| 备选路径 | 2a、用户将要查询的网络图片地址填在系统页面的检索栏，提交图像检索请求。<br>3a、用户提交的文件格式不正确。 |

| | |
|---|---|
| | 1、系统提示用户提交的文件格式不正确，不接收用户的检索请求。<br>2、返回第 2 步。<br>4a、系统不能正确获取图片流。<br>1、系统提示用户获取图片失败，不接收检索请求。<br>2、返回第 2 步。<br>6a、获取检索结果失败或者检索结果为空。<br>1、系统提示用户未能找到相似结果。<br>2、完成。 |
| 后置条件 | 用户请求成功提交到服务端，服务端进行了正确的处理并向用户返回检索结果 |
| 特殊需求 | 网络稳定，浏览器支持 JavaScript |

使用活动图对上述用例主要业务流程做进一步描述，如图 3-8。

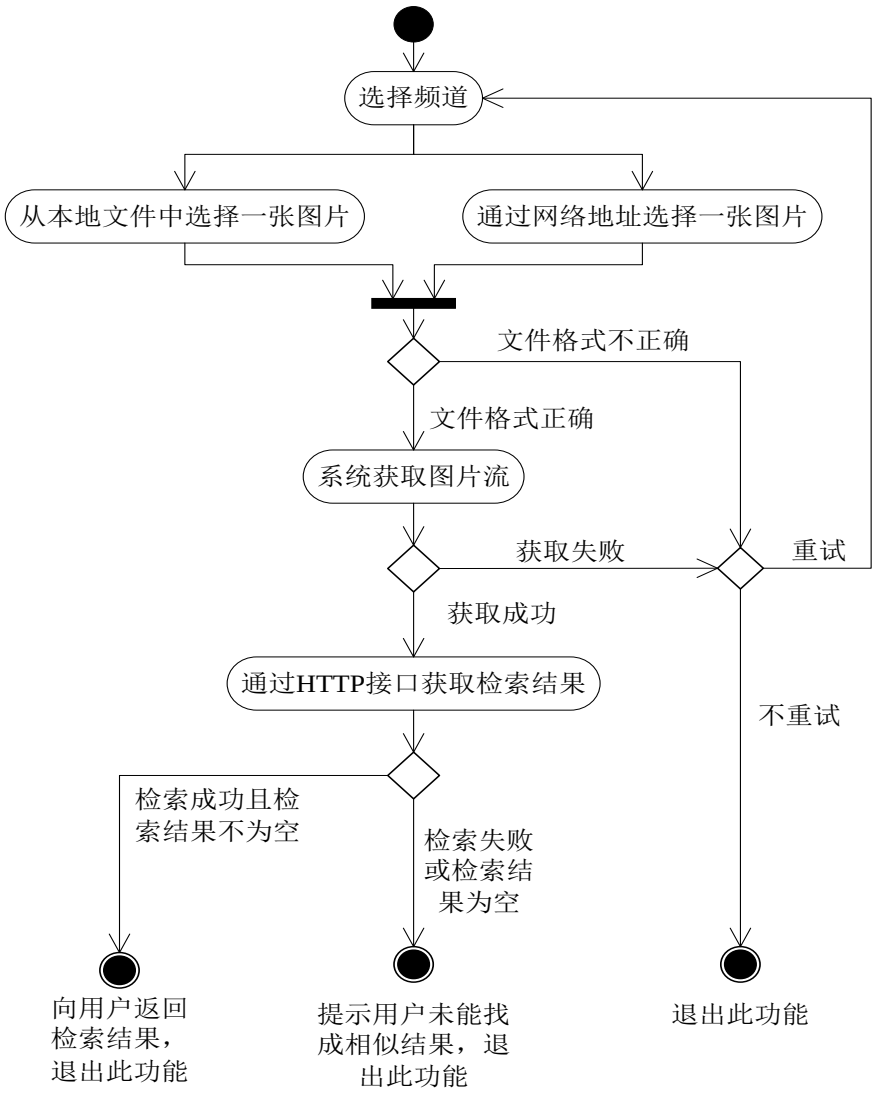



图 3-8：图片检索用例活动图

### 3.2.3 手机客户端

在本小节中所说的用户都是指搜索用户。用户在使用系统提供的手机客户端时可能进行的操作有：1、使用系统的拍照搜索功能；2、使用系统的图片搜索功能；3、查看搜索结果中其中一项的详细信息。对用户的操作进行分析得到三个基本用例：拍照搜索、图片搜索、查看结果详情。用例图如图 3-9 所示。

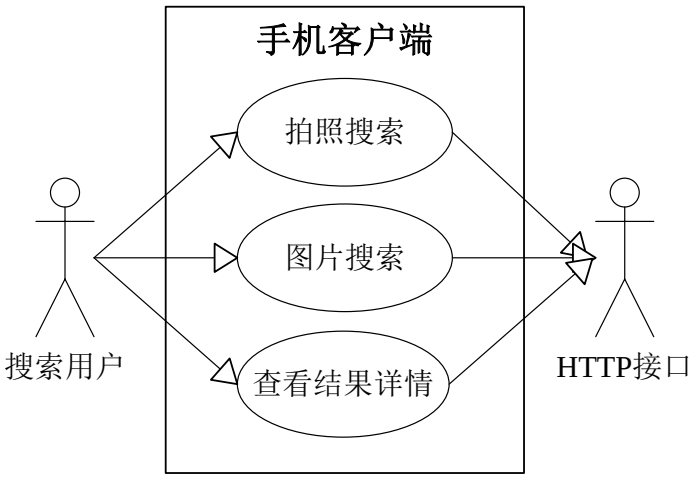



图 3-9：手机客户端用例图

本小节中的图片搜索和查看结果详情用例与 3.2.2 小节中的图片检索和查看结果详情用例基本一致，在本小节中不再进行分析。表 3-7 对拍照搜索用例进行详细描述。

表 3-7：拍照搜索用例详述

| | |
|---|---|
| 用例名 | 拍照搜索 |
| 用例概述 | 用户通过手机相机拍照，并利用系统的可视化检索功能来检索相似图片 |
| 使用频率 | 高 |
| 主要参与者 | 用户 |
| 前置条件 | 用户打开 iSearch 手机客户端，并选择拍照搜索 |
| 基本事件流 | 1、 用户使用手机拍摄一张要进行搜索的照片。<br>2、 用户提交搜索请求。<br>3、 手机客户端通过 Http 接口从 iSimilar 平台获取检索结果。<br>4、 客户端向用户输出查询的图片及查询结果。<br>5、 完成。 |
| 备选路径 | 2a、用户取消搜索请求。<br>1、客户端不向 Http 接口发送请求。<br>2、返回上一页。<br>2b、用户选择重新拍摄一张照片。<br>1、客户端重新调用手机相机的拍照功能。<br>2、返回第 1 步。 |
| 后置条件 | 用户成功拍摄照片，并且通过 http 请求成功获取检索结果 |
| 特殊需求 | 手机支持联网 |

### 3.2.4 广告注册模块

本小节中所说的用户均为广告用户。用户在使用系统提供的广告注册模块时可能进行的操作有：1、登录系统；2、注册一个新的账户；3、已注册的用户对自己的账户信息进行修改；4、退出系统；5、向系统上传标注了扩展信息的广告图片；6、用户对自己上传的广告进行管理。对用户的操作进行分析得到六个基本用例：用户登录、用户注册、退出系统、修改账户信息、上传广告、广告管理。用例图如图 3-10 所示。

在本小节中，用户登录、退出系统及修改账户信息用例使用摘要形式描述，用户注册、上传广告及广告管理用例采用详述形式描述。

**摘要用例：用户登录**

用户通过浏览器访问系统登录页面，根据提示输入用户名及密码并提交登录请求。系统接收请求并判断用户名密码是否正确。正确则进入广告注册页面，否则返回错误提示要求用户重新输入用户名密码。

**摘要用例：退出系统**

用户希望安全退出系统，点击页面上的“退出”链接或按钮，提交退出系统请求。系统接收请求，将相应的账户状态修改为未登录，返回首页。

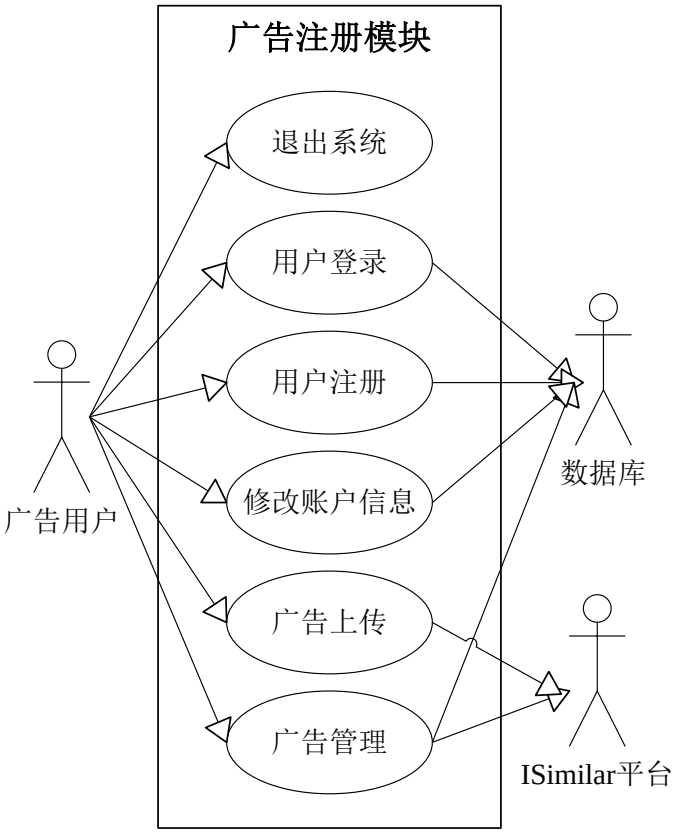



图 3-10：广告注册模块用例图

**摘要用例：修改账户信息**

用户选择“账户管理”操作，根据提示输入新的密码、确认密码及简介并提交修改请求。系统判断两次输入的密码是否一致。一致则将对相应的账户信息进行修改；否则否则返回错误提示要求用户重新检查密码。

表 3-8：用户注册用例详述

| | |
|---|---|
| 用例名 | 用户注册 |
| 用例概述 | 用户注册一个账户，以使用系统的广告上传功能 |
| 使用频率 | 中，用户根据需要选择使用 |
| 主要参与者 | 用户 |
| 前置条件 | 用户通过浏览器访问系统的注册页面 |
| 基本事件流 | 1、 用户根据提示填写账户名、密码、确认密码及邮箱地址，并提交注册请求。<br>2、 系统检查用户输入并确认必填项填写完整且正确。<br>3、 系统接收用户注册请求并检查要注册信息是否合法。<br>4、 信息合法，系统新增一条账户信息，注册成功。<br>5、 完成。 |
| 备选路径 | 2a、两次输入的密码不一致。<br>1、系统提示用户检查密码，不接收注册请求。<br>2、返回第 1 步。<br>2b、邮箱地址不正确。<br>1、系统提示用户检查邮箱地址，不接收注册请求。<br>2、返回第 1 步。<br>4a、请求注册的用户名已存在。<br>1、系统提示用户账户名已被注册，不新增账户信息。<br>2、返回第 1 步。<br>4b、注册信息中的邮箱地址已被使用。<br>1、系统提示用户邮箱地址已被使用，不新增账户信息。<br>2、返回第 1 步。 |
| 后置条件 | 用户注册成功，可通过注册的用户名和密码登录系统 |
| 特殊需求 | 网络稳定，浏览器支持 JavaScript |

表 3-9：上传广告用例详述

| | |
|---|---|
| 用例名 | 上传广告 |
| 用例概述 | 用户向系统提交标注了相关扩展信息的广告图片 |
| 使用频率 | 中，根据用户需要选择使用 |
| 主要参与者 | 用户 |
| 前置条件 | 用户登录系统并进入广告上传页面 |
| 基本事件流 | 1、 用户从本地计算机中选择要上传的图片，点击查看图片。<br>2、 系统确认用户提交的文件格式正确。<br>3、 系统获取用户提交的图片流并输出到页面。<br>4、 用户根据提示填写相应信息。<br>5、 填写完成后，用户提交上传广告请求。<br>6、 系统检查用户输入并确认必填项填写完整且正确。<br>7、 系统接收用户请求并将数据写入 iSimilar 平台的图片仓库。<br>8、 系统向用户返回操作成功提示。<br>9、 完成。 |

| | |
|---|---|
| 备选路径 | 1a、用户将要查询的网络图片地址填在系统页面的检索栏，提交图像检索请求。<br>2a、用户提交的文件格式不正确。<br>1、系统提示用户提交的文件格式不正确。<br>2、返回第 1 步。<br>3a、系统不能正确获取图片流。<br>1、系统提示用户获取图片失败。<br>2、返回第 1 步。<br>5a、用户取消上传广告，完成。<br>6a、必填项未填写完整或不正确。<br>1、系统向用户返回相应提示，不接收上传请求。<br>2、返回第 4 步。<br>8a、存储过程中系统发生异常。<br>1、系统向用户返回相应提示。<br>2、返回第 5 步。 |
| 后置条件 | 用户上传广告及相关扩展信息成功 |
| 特殊需求 | 网络稳定，浏览器支持 JavaScript |

使用活动图对上述用例主要业务流程做进一步描述，如图 3-11。

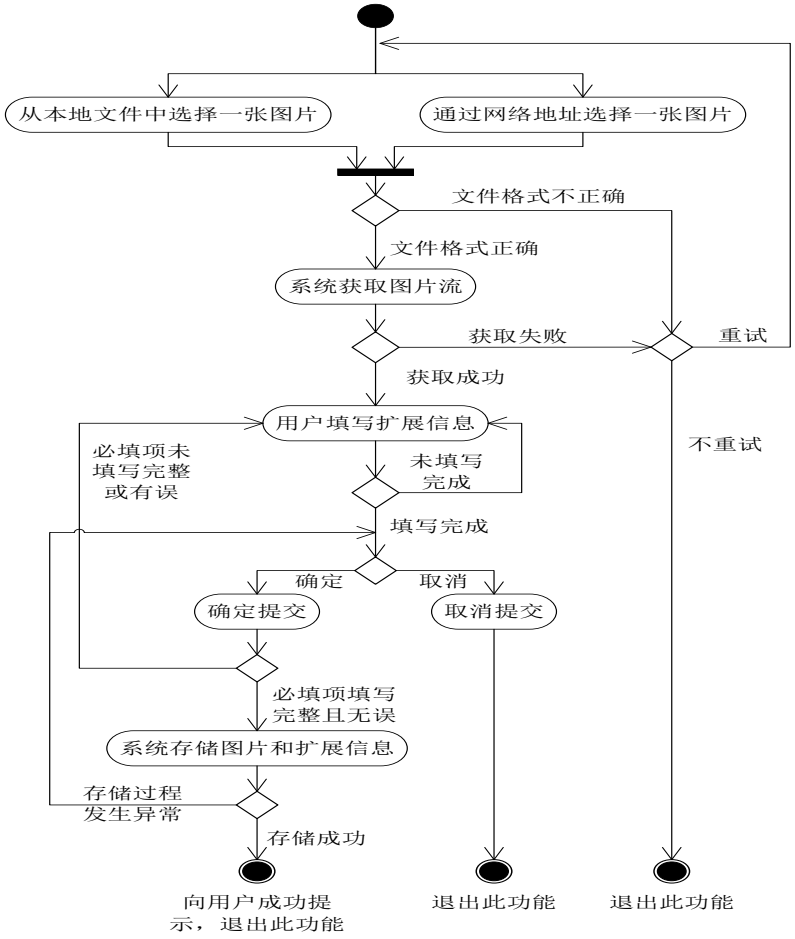



图 3-11：上传广告用例活动图

表 3-10：广告管理用例详述

| | |
|---|---|
| 用例名 | 广告管理 |
| 用例概述 | 用户对自己账户下已上传的广告信息进行管理，包括修改、删除等操作 |
| 使用频率 | 低，用户根据需要选择使用 |
| 主要参与者 | 用户 |
| 前置条件 | 用户登录系统并进入广告管理页面 |
| 基本事件流 | 1、 用户浏览本账户上传的广告列表。<br>2、 用户从列表中选择一项进行查看。<br>3、 系统接收请求并根据用户选择返回相应的广告信息。<br>4、 用户对已上传的广告的相关扩展信息进行修改（修改过程与上传广告用例中的基本事件流步骤 4 至 8 一致）。<br>5、 完成。 |
| 备选路径 | 3a、用户请求查看的广告信息不存在。<br>1、系统提示用户请求查看的广告信息不存在或已被删除。<br>2、返回第 1 步。<br>4a、用户提交删除广告请求。<br>1a、系统接收请求并删除相应的广告信息。<br>1b、请求删除的广告信息不存在，系统向用户返回相应提示。<br>2、刷新广告列表。<br>3、返回第 1 步。 |
| 后置条件 | 用户请求成功提交到服务器并得到正确处理 |
| 特殊需求 | 网络稳定，浏览器支持 JavaScript |

## 3.3 iSearch 系统领域分析和建模

通过对用例的分析，可以确定系统边界、主要参与者及主要业务流程。业务流程中出现的名词包括了业务的行为主体及业务过程中的操作实体，对这些名词进行进一步分析有助于抽象业务模型，发现概念类[14]，并建立领域模型。

从用例中可以看出，系统的用户主要有三类，并且各类用户所使用的功能模块不同：搜索用户使用系统 Web 端和手机客户端提供的可视化图像检索功能；广告注册模块主要面向广告用户；而数据操作模块的使用者是系统用户。下面按用户分类来对系统进行领域模型的分析、设计和建模。

对主要参与者是搜索用户的用例模型进行分析，得到以下关键词：1、搜索用户、系统、手机客户端；2、iSimilar 平台、文件系统；3、Http 接口、图片流、检索结果；4、

[14]概念类是思想、事物或对象，可以从其符号、内涵和外延来考虑。参考自《UML 和模式应用》第 3 版，Craig Larman 著。

页面、相机、本地文件、请求。其中第一组是参与交互的对象，第二组是外部参与者，第四组是属性。对关键词进行分析和筛选后得到领域模型如图 3-12。

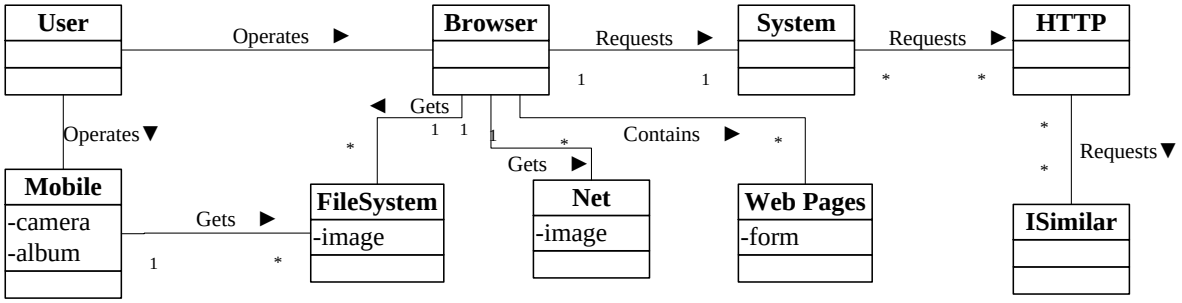



图 3-12：面向搜索用户部分领域模型图

对主要参与者是广告用户的用例模型进行分析，得到以下关键词：广告用户、系统、账户管理模块、账户信息、账户名、密码、邮箱地址、简介、请求、iSimilar 平台、页面、数据库、广告列表、广告图片、扩展信息。经过分析和筛选后得到领域模型如图 3-13。

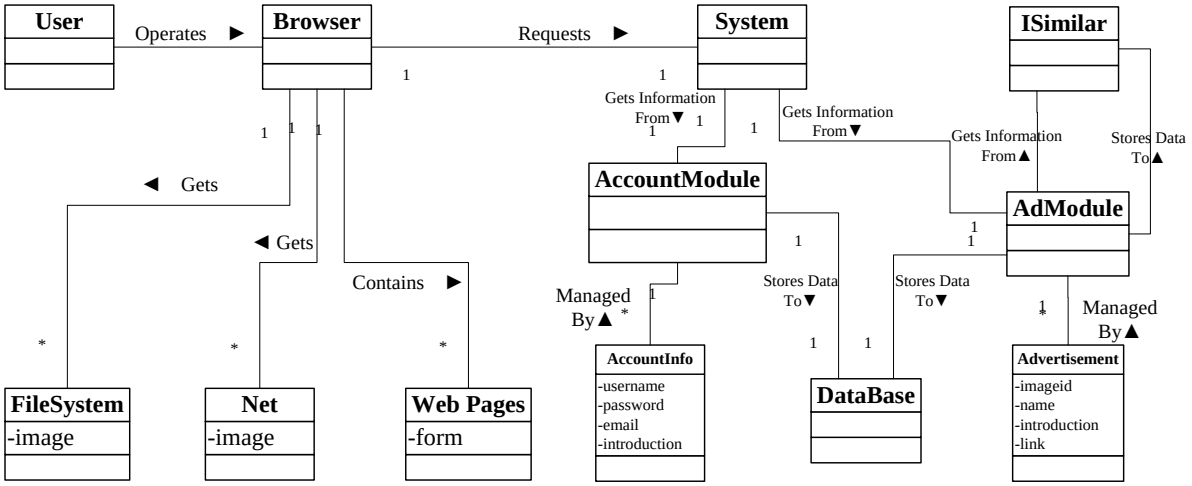



图 3-13：面向广告用户部分领域模型图

对主要参与者是系统用户的用例模型进行分析，得到以下关键词：系统用户、系统、命令行终端、爬虫、插件、爬虫配置、种子文件、抓取任务、任务 ID、任务状态、命令、信息提取方法、模板、请求、页面、测试结果、提取结果、HDFS、Hadoop、统计结果。经过分析和筛选后建立领域模型如图 3-14 所示。

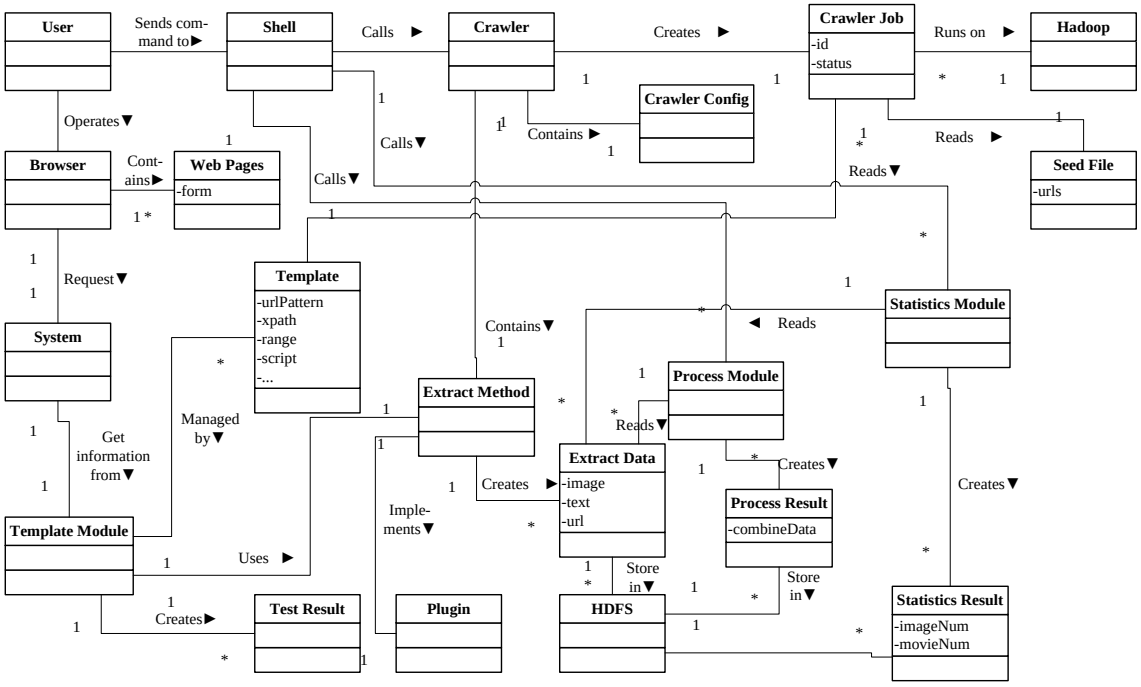



图 3-14：面向系统用户部分领域模型图

## 3.4 iSearch 系统的其他需求

iSearch 系统除了需要满足前文用例分析中所体现的功能需求外，还应用满足以下需求：

（1） 性能

这里提到的性能需求包括两个方面：一方面是系统的响应时间，在网络条件良好的情况下，用户提交图像检索请求对服务器返回检索结果之间的用户等待时间应该不超过 20 秒，广告上传的等待时间应该不超过 10 秒，其他如登录、注册等操作的等待时间不超过 5 秒；另一方面是网络爬虫的抓取效率，要根据抓取任务的状态，及时调整配置，充分利用带宽，并加快内容提取的处理速度。

（2） 可靠性

Hadoop 的数据复制机制使得一个数据磁盘的损坏不会造成数据的丢失及系统的崩溃，提高了系统的可靠性。并且系统的崩溃不会对已完成处理的数据造成影响。只需重新启动系统即可继续进行操作。

（3） 易用性

系统为搜索用户和广告用户提供简单方便的操作，并且系统提供明确的操作提示。搜索用户及广告用户无需进行培训即可正确地使用系统。系统用户要求具有一定的计算机及 Hadoop 背景知识。经过一天培训的新的系统用户应该能够正确使用数据操作模块80%以上的功能。

（4） 容错性

系统对用户操作不当造成的错误或系统内部异常有相应的处理机制，尽可能地避免系统崩溃，以提高系统的可用性。

（5） 安全性

系统为不同类型的用户提供了不同的功能及操作权限，一定程度上提高了系统的安全性，并避免了由于越权操作造成的数据泄露、丢失和损坏。

（6） 可移植性

系统采用 JAVA 语言开发，可以移植到支持 JAVA 运行环境的平台上。数据操作模块还额外要求系统安装配置 Hadoop 运行环境。

（7） 可扩展性

通过网络爬虫 Nutch 提供的插件机制，可以方便地对网络爬虫进行功能扩展。

（8） 可维护性

系统各模块间基本独立，对一个模块进行修改不会影响其他模块的使用，一定程度上方便了系统的维护。另外，Web 应用程序采用 SSH 框架搭建，结构清晰，表示层、业务逻辑层和数据持久层之间耦合度小，各层内的变化对其他层的影响较小，提高了系统的可复用性及开发效率。

# 第四章　iSearch 系统架构设计

在第三章中笔者对 iSearch 系统的需求进行了简单分析，从本章开始对系统的设计进行讨论。

本章从架构的角度来对系统设计进行分析。4.1 小节主要对 iSearch 系统的架构及原理进行了介绍。4.2 小节将对第三章中数据处理模块的“模板修改”及“模板文件测试”、Web 可视化检索模块的“图片检索”和广告注册模块的“广告上传”用例进行业务流程实现。4.3 至 4.5 小节分别对获取图像检索结果的 Http 接口设计、增强索引设计及删除广告解决方案进行了介绍。4.6 小节对 iSearch 系统的数据库设计进行介绍。4.7 小节则给出了系统的出错信息及相应处理策略。

## 4.1　iSearch 系统架构及原理

iSearch 系统有三个交互接口：浏览器、手机、Linux 系统命令行终端。这意味着 iSearch 系统是由 Web 应用程序、手机应用程序及 JAVA 可执行程序组成而成的。系统的整体架构设计见图 4-1。

下面根据图 4-1 对系统的架构进行简单介绍。

手机应用程序部分，系统遵循 Android 及 Windows Phone 手机应用程序开发的架构进行设计。用户可通过利用手机的拍照功能拍摄照片或从手机相册选择图片作为检索图片并向系统提交检索请求，通过系统封装的 Http 接口从 iSimilar 平台及 MySQL 数据库获取检索结果，应用程序接收结果并分析处理后输出到手机界面。

JAVA 可执行程序部分，包括了扩展的 Nutch 网络爬虫，数据分析及处理部件，和数据统计部件。网络爬虫及数据分析处理部件均采用 MapReduce 分布式框架进行设计。网络爬虫部分以开源分布式爬虫 Nutch 为基础进行网页的数据抓取及 URL 的调度，并通过扩展的本论文中提出的网页提取方法对抓取的数据进行分析和抽取，数据存储格式均为 key/value 对。数据分析及处理部件根据处理数据不同会有不同的实现方式，大概流程为：1、提取 value 中的关联标记，将其作为新的 key 并从 value 中删除；2、将拥有相同 key 的所有 value 合并；3、重复步骤 1 和 2，直至 value 中不再有关联标记，即所

有关联的数据都已经合并到一起。这些部分的实现将在 5.2 小节中做进一步介绍。

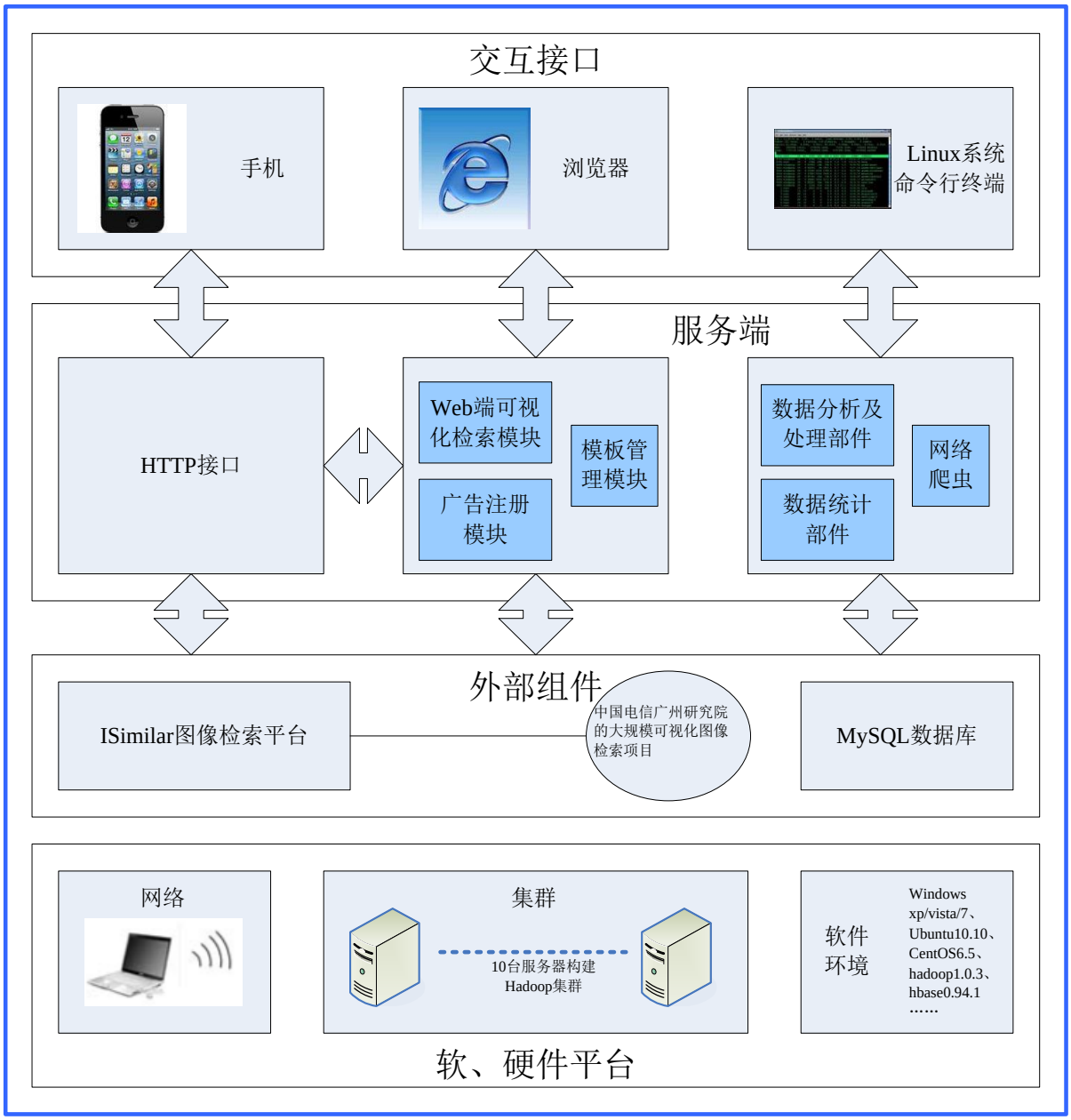



图 4-1：iSearch 系统整体架构图

Web 应用程序部分采用 B/S 结构，根据面向的用户类型及需求不同又分为两部分：面向搜索用户及广告用户的部分使用了 SSH 框架进行构建；面向系统用户的部分出于需求变动小且不涉及数据库操作的考虑，没有使用框架进行设计。图 4-2 展示了 Web 应用程序部分的简单架构图。

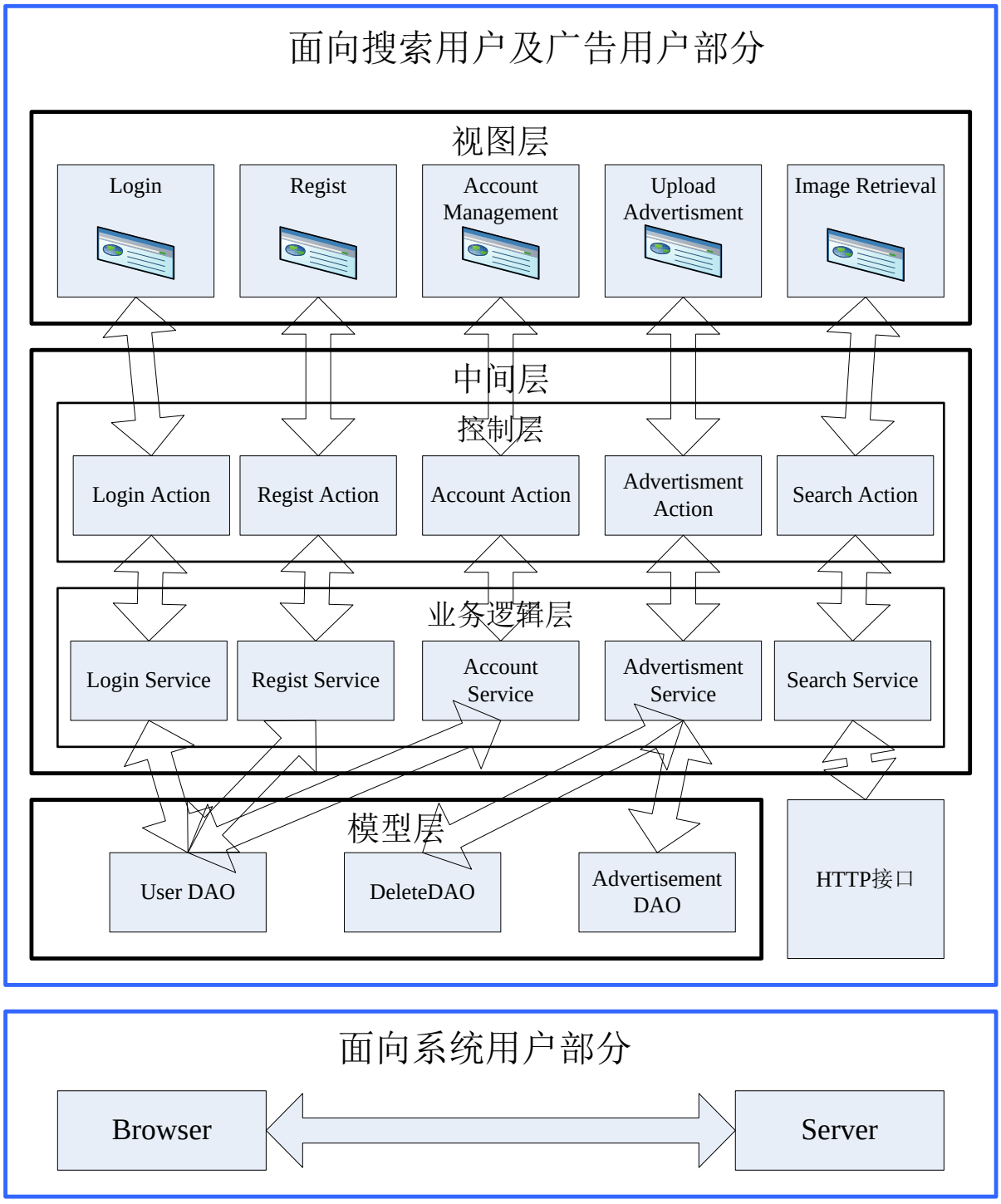



图 4-2：iSearch 系统 Web 部件架构图

面向搜索用户及广告用户的 Web 应用程序部分使用 SSH 框架构建，分为视图层、中间层及数据模型层，而中间层又可细分为控制层和业务逻辑层。视图层是用户与系统之间交互的接口，提供了数据输出及用户操作的方式；从图 4-2 中可以看到，该部分的视图层提供的操作有：登录、注册、账户信息管理、广告上传和图像检索。控制层负责

控制业务逻辑层与视图层的交互，它拦截所有的 HTTP 请求，并根据用户请求调用相应的业务逻辑进行处理，然后根据处理的结果选择相应的视图返回给用户；该部分的控制层为视图层中的操作分别提供了相应的控制器。业务逻辑层负责处理用户请求，实现业务逻辑，它通过调用模型层来完成数据处理；业务逻辑层对应控制层分为了 5 个部分：（1）登录，包括账户名密码检查及控制单点登录；（2）注册，负责创建新账户并避免注册账户重复；（3）账户信息管理，提供修改密码及简介的功能，和向登录用户提供对本账户已上传广告的修改和删除功能；（4）广告上传，向登录用户提供向系统提交广告图片及相关扩展信息的功能。（5）图像检索，向用户提供电影、衣服、广告三个频道的可视化图像检索功能。模型层负责与持久化对象交互，完成数据处理；对应的数据库设计在 4.6 小节中进行介绍。

面向系统用户部分的前台界面采用 JSP 来实现。JSP 文件在传统网页文件中加入 JAVA 代码及 JSP 标记，服务器在接收到访问 JSP 页面的请求时，会执行页面中的 JAVA 代码，然后根据执行结果生成 HTML 页面返回到浏览器，实现了页面数据的动态显示。程序操作都是在服务器端完成的，浏览器端得到的实际上是处理后的 HTML 文本，因此访问 JSP 网页并不需要浏览器对 JAVA 的支持。后台部分使用 JAVA 语言实现，提供了在 HDFS 上新建、修改、删除模板的操作，还提供了利用本论文提出的网页信息提取访求对模板进行测试的功能。

在 iSearch 系统中，Web 应用程序部分的界面除了使用 JSP 动态网页技术外，还使用了 css 和 Ajax 改善页面布局及系统响应方式，为用户提供更良好的使用体验。

## 4.2 iSearch 系统重要业务用例实现

上一节给出了 iSearch 系统的整体架构设计，并对各部分的功能及联系进行了清楚的说明。在本小节中，将对第三章中数据处理模块的“修改模板文件”及“模板文件测试”、Web 可视化检索模块的“图片检索”和广告注册模块的“广告上传”这几个关键用例进行分析和实现，对三类用户与系统之间的交互进行进一步探讨。

### 4.2.1 修改模板文件及模板文件测试用例实现

在本小节中，将“修改模板文件”及“模板文件测试”这两个用例综合起来进行讨论。顺序图见图 4-3。

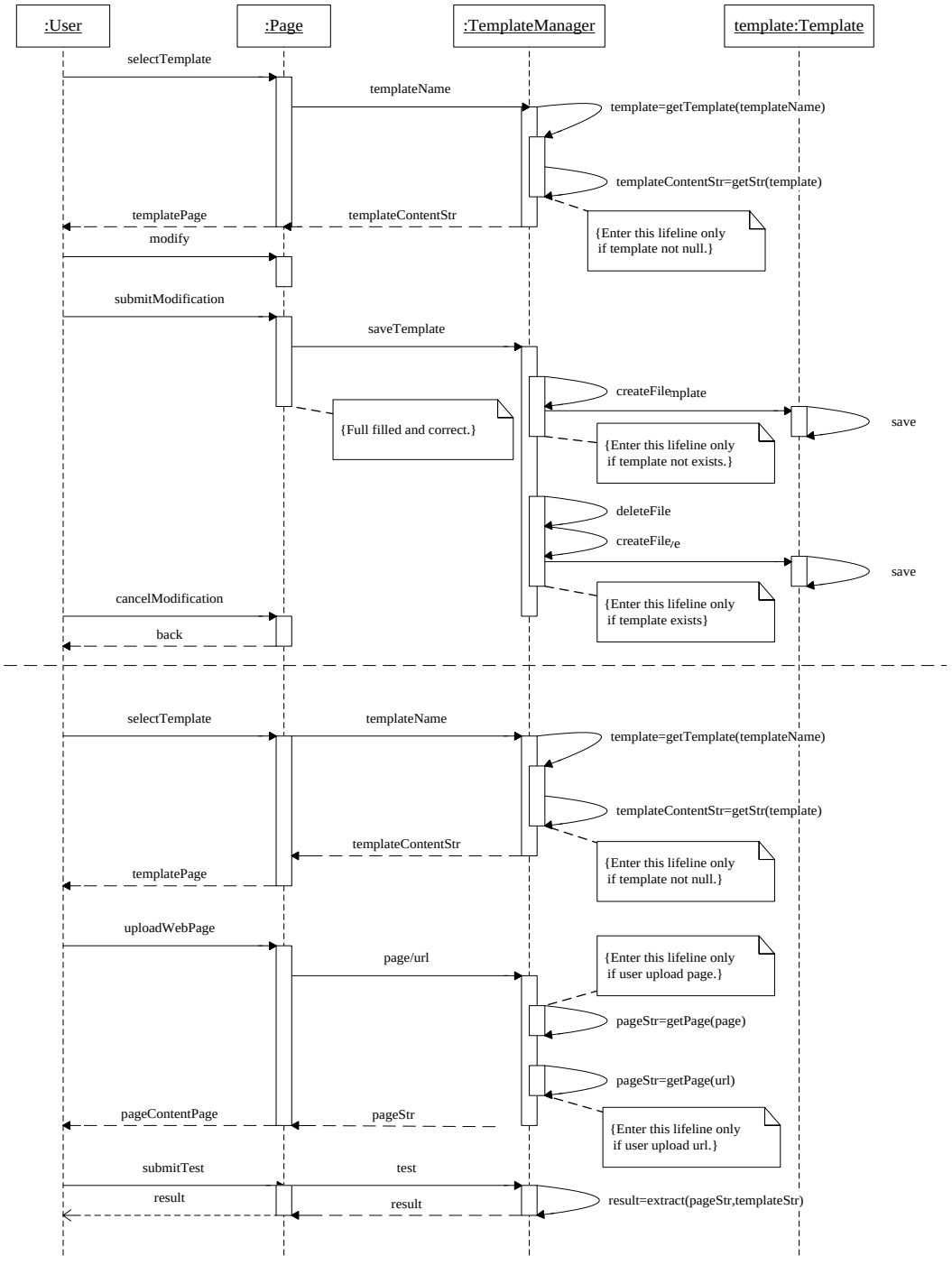



图 4-3：修改模板文件及模板文件测试顺序图

修改模板文件及模板文件测试两个用例的用户与系统之间的交互和系统内的交互过程如下：

（1）用户 User 从浏览器当前页面 Page 的模板列表中选择一项（selectTemplate），并向模板管理模块（TemplateManager）提交修改模板请求。

（2）模板管理模块接收请求并获取相应的模板（getTemplate），将模板内容转为字符串格式（templateContentStr）并返回。若获取的模板为空，则返回空值。

（3）用户根据需求在页面 Page 上修改模板（modify）。

（4）用户完成修改操作后向模板管理模块提交确认修改请求（submitModification）。若用户取消修改（cancelModification），返回模板列表页面。

（5）经检查输入完整且无误，模板管理模块根据用户输入进行相应的修改（saveTemplate），否则返回错误提示。

（6）若模板文件不存在，新建文件并保存模板内容。若模板文件已存在，删除原有文件，新建文件并保存。

（7）用户从模板列表页面（Page）的列表中选择一项（selectTemplate），模板管理模块向用户返回模板内容字符串。

（8）用户向模板管理模块上传用于测试模板文件的文件或 URL（uploadWebPage）。若是直接上传文件，模板管理模块直接读取文件内容（getPage(page)）；若用户提交的是网页地址，模板管理模块通过地址来获取文件内容（getPage(url)）。向用户返回文件内容。

（9）用户提交测试模板文件请求（submitTest），模板管理模块接收请求并调用相应的方法进行测试（extract），并将测试结果（result）返回给用户。

### 4.2.2 图像检索用例实现

图像检索用例的顺序图如图 4-4 所示。

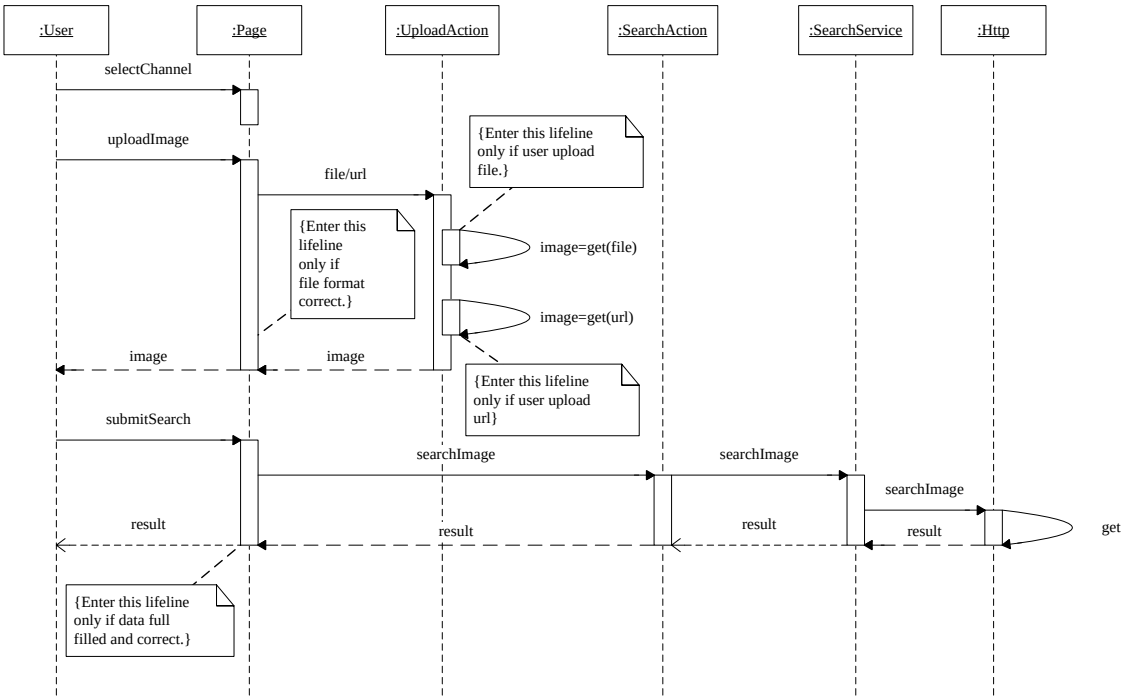



图 4-4：图像检索顺序图

图像检索用例的用户与系统之间的交互和系统内的交互如下：

（1） 用户（User）通过浏览器选择要使用的检索频道（selectChannel）。

（2） 用户（User）通过浏览器当前页面（Page）向系统上传一张图片，系统检查图片格式并确认格式正确，否则返回错误提示。

（3） 若用户上传的是本地图片，系统直接获取图片流（get(file)）；若用户提交的是图片网络地址，系统通过地址获取图片数据（get(url)）；向用户返回图片。

（4） 用户提交检索请求（submitSearch），系统接收请求并通知检索部件（SearchAction）进行处理。

（5） 检索部件通知检索业务逻辑处理部件（SearchService）进行业务处理（searchImage）。

（6） 业务逻辑处理部件通过 HTTP 接口获取检索结果（result），并返回给用户。

### 4.2.3 广告上传用例实现

广告上传用例的顺序图如图 4-5 所示。

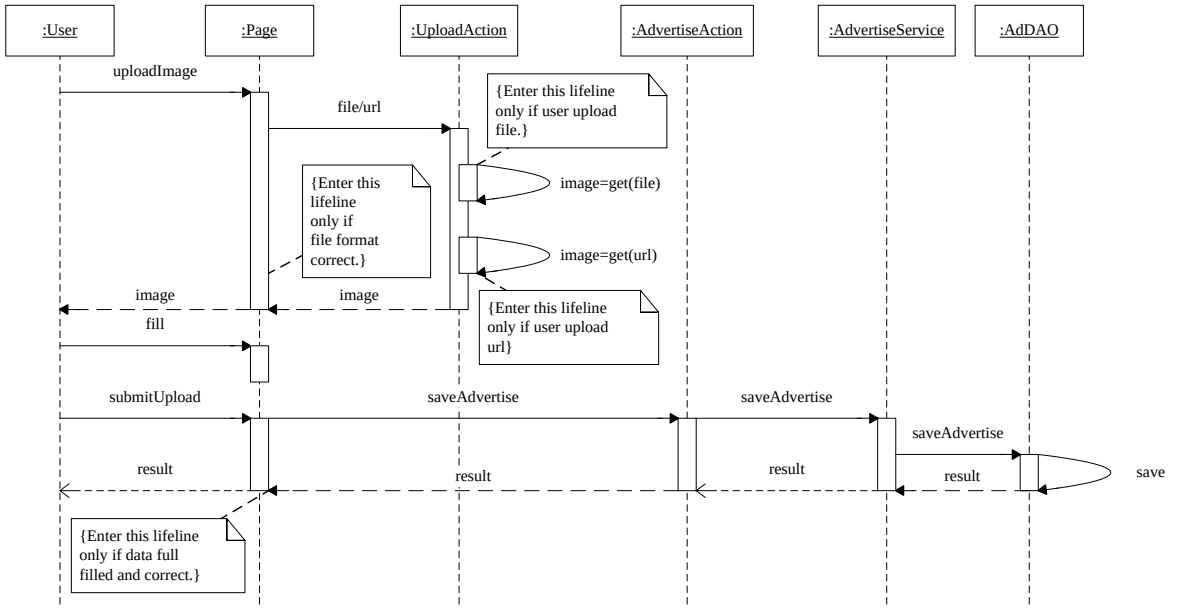



图 4-5：广告上传顺序图

广告上传用例的用户与系统之间的交互和系统内的交互过程如下：

（1） 用户（User）通过浏览器当前页面（Page）向系统上传一张图片，系统检查图片格式并确认格式正确，否则返回错误提示。

（2） 若用户上传的是本地图片，系统直接获取图片流（get(file)）；若用户提交的是图片网络地址，系统通过地址获取图片数据（get(url)）；向用户输出图片。

（3） 用户根据需要填写扩展信息（fill），填写完成后提交上请求（submit）。

（4） 系统检查输入并确认必填项填写完整且无误，然后将请求发送到广告处理部件（AdvertiseAction）；否则返回错误提示。

（5） 广告处理部件通知广告业务逻辑部件（AdvertiseService）对上传的数据进行存储。

（6） 广告业务逻辑部件通过调用广告模型访问部件（AdDAO）完成数据处理和存储（saveAdvertise），并返回处理结果。

将图片存储到图片仓库后，需要进行增量索引操作。**增量索引方法是本论文的主要技术贡献之一。**图 4-6 给出了本论文中所使用的增量索引方法的流程图。

增量索引方法的处理流程如下：

（1） 系统接收用户提取的图片流并存储到 iSimilar 平台的图片仓库，获取返回的图片 ID。

（2） 系统调用增量索引部件，索引部件根据配置进行初始化，包括特征提取方法和特征映射方法的选择。

（3） 特征提取器提取图片特征。

（4） 分析器获取特征提取结果并进行映射，结果为<分析器名，映射值>的键/值对，可使用多个分析器。

（5） 根据分析结果构建Lucene的索引单元 Document（Document 中的 field 为一个索引字段，name 为索引字段名，对应分析器名；value 为要索引的数据，对应映射值；还需要将图片 ID 也写入 Document 中，以便获取实际的图片数据）。

（6） 新建增量索引文件并将索引数据写入。

（7） 若增量索引文件数量达到 M（M 为系统用户配置），则执行合并索引文件操作。合并后的索引文件数量由索引数据量及索引文件大小决定，例如索引数据量为 60M，而规定的索引文件大小限制为 16M，则索引文件数为 4。

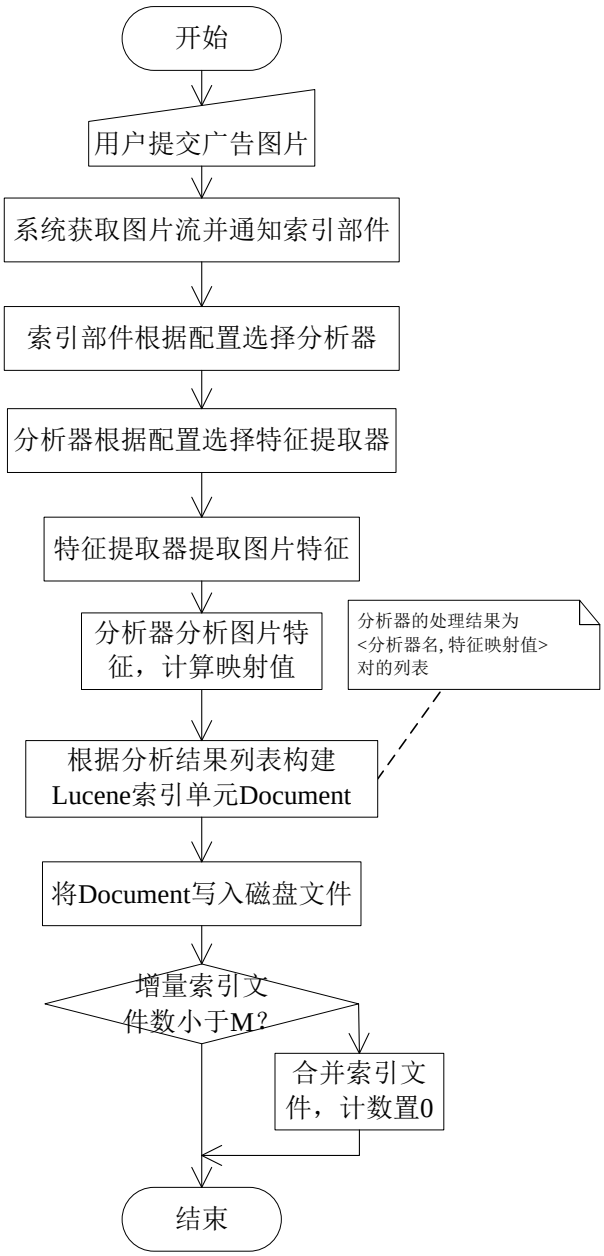



图 4-6：增量索引方法流程图

## 4.3 Http 接口概要设计

本论文设计并实现了 iSearch 系统图像检索功能的 HTTP 接口，可以方便地通过 HTTP 协议获取格式化的图像检索结果。HTTP 接口的业务处理过程如图 4-7 所示。

HTTP 接口处理检索请求及获取详情请求，过程如下：

（1） 处理检索请求

1、分析 HTTP 协议，获取要检索的图片流及检索参数（参数包括要检索的图片仓库名、标记名，起始点，要获取的结果数等）。

2、根据参数选择选择要检索的图片仓库，初始化检索请求。

3、调用 iSimilar 平台的检索方法，获取检索结果。

4、对检索结果进行分析和处理，并封装为 JSON 格式以便客户端读取。

5、客户端通过图片 ID 提交 HTTP 请求获取相应的图片数据。

6、根据用户请求从相应的图片仓库中获取图片数据并返回给用户。

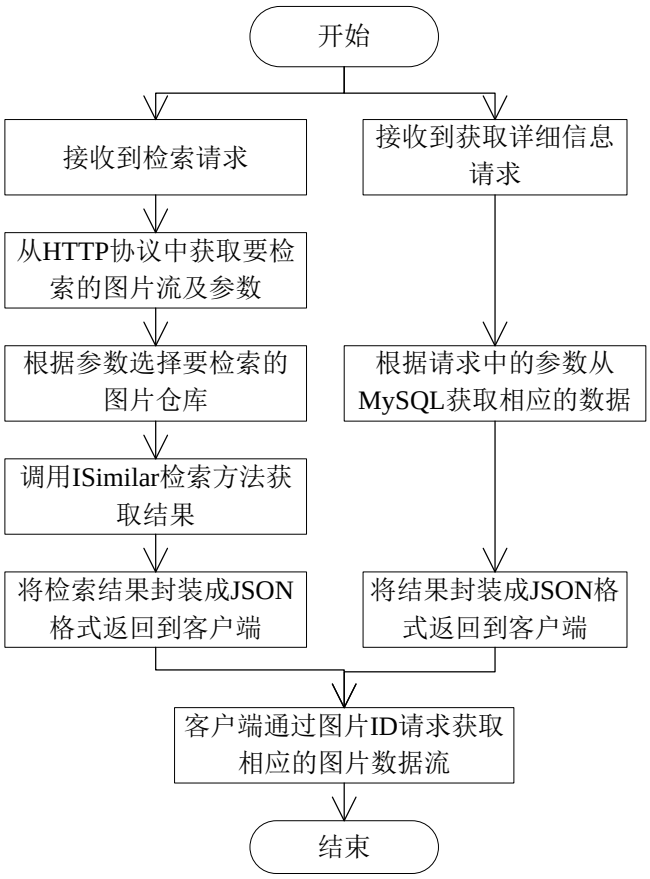



图 4-7：HTTP 接口业务处理过程

（2） 处理获取详情请求

1、分析 HTTP 协议，获取参数（包括图片 ID、图片来源 URL 等）。

2、根据参数从 MySQL 数据库中获取相应的数据（包括详细介绍、相关图片等）。

3、将结果封装成 JSON 格式以便客户端处理。

4、客户端通过图片 ID 提供 HTTP 请求获取图片数据。

5、根据图片 ID 返回相应的图片数据。

## 4.4 删除广告方法概要设计

iSimilar 图片仓库没有提供对库内数据的删除功能，不能满足 iSearch 系统删除广告数据的需求，本论文在不改动 iSimilar 图片仓库核心架构的前提下，为这个问题提出了预选的解决方案：记录删除的图片 ID，在向用户返回检索结果时根据删除记录将相应

的图片ID从结果中剔除。这个方案并没有真正地将图片数据从iSimilar图片仓库中删除，只是简单地对检索结果进行了处理，使得从用户得到的检索结果来看该数据项已被删除。这个方法存在一个很大的弊端：当删除的数据越来越多时，可能造成某次检索结果中包含大量“已删除”图片的 ID，导致用户得到的检索结果减少，降低用户体验。这是本论文未来工作的重点之一。

## 4.5 数据库设计

iSearch 系统中图片的存储和获取通过 iSimilar 平台提供的接口来完成，不涉及数据库操作。iSimilar 平台的图片存储仓库虽然提供了对文本数据的存储方法，但在文本数据过多、过长时会对 iSimilar 平台的索引性能造成影响。因此，在本论文中，仅将图片数据存储到 iSimilar 的图片仓库，文本数据则是通过 MySQL 数据库进行存储和管理，图片及相应的文本数据通过图片 id 号进行关联。

下面给出系统中所使用的数据表的详细设计。

表 4-1：账户信息表

| **表名** | **t_user** | | | |
|---|---|---|---|---|
| **字段名** | **数据类型** | **长度** | **约束** | **描述** |
| username | varchar | 20 | 非空、主键 | 用户名 |
| password | varchar | 20 | 非空 | 密码 |
| email | varchar | 50 | 非空 | 邮箱地址 |
| intro | tinytext | 最大长度 65535 字节 | 无 | 账户简介 |

表 4-2：删除广告图片 ID 表

| **表名** | **t_delete** | | | |
|---|---|---|---|---|
| **字段名** | **数据类型** | **长度** | **约束** | **描述** |
| id | varchar | 30 | 非空、主键 | 删除的图片 ID |

表 4-3：广告数据表

| **表名** | **t_ad** | | | |
|---|---|---|---|---|
| **字段名** | **数据类型** | **长度** | **约束** | **描述** |
| id | varchar | 30 | 非空、主键 | 图片 ID |
| username | varchar | 20 | 非空 | 用户名 |
| name | varchar | 100 | 非空 | 图片名 |
| link | varchar | 300 | 无 | 相关链接 |
| intro | text | 最大长度 65535 字节 | 无 | 广告图片简介 |

表 4-4：时光网电影详细信息表

| 表名 | t_mtime | | | |
|---|---|---|---|---|
| 字段名 | 数据类型 | 长度 | 约束 | 描述 |
| url | varchar | 255 | 非空、主键 | 时光网电影网址 |
| theater | varchar | 255 | 无 | 影院信息网址 |
| name | text | 最大长度 65535 字节 | 非空 | 电影名 |
| director | text | 最大长度 65535 字节 | 无 | 导演 |
| actor | text | 最大长度 65535 字节 | 无 | 演员 |
| type | text | 最大长度 65535 字节 | 无 | 电影类型 |
| area | text | 最大长度 65535 字节 | 无 | 所属地区 |
| time | text | 最大长度 65535 字节 | 无 | 上映时间 |
| language | text | 最大长度 65535 字节 | 无 | 语言 |
| duration | text | 最大长度 65535 字节 | 无 | 片长 |
| nickname | text | 最大长度 65535 字节 | 无 | 别名 |
| plot | longtext | 最大长度 4294967295 字节 | 无 | 剧情 |

表 4-5：电影影评表

| 表名 | t_mtimecomm | | | |
|---|---|---|---|---|
| 字段名 | 数据类型 | 长度 | 约束 | 描述 |
| id | int | 11 | 非空、主键 | 自增主键 |
| url | varchar | 255 | 非空 | 电影网址 |
| title | tinytext | 最大长度 255 字节 | 无 | 影评标题 |
| user | varchar | 50 | 非空 | 用户名 |
| time | varchar | 50 | 非空 | 评论时间 |
| comment | text | 最大长度 65535 字节 | 无 | 影评内容 |
| link | varchar | 255 | 无 | 影评全文链接 |

表 4-6：电影短评表

| 表名 | t_mtimeshortcomm | | | |
|---|---|---|---|---|
| 字段名 | 数据类型 | 长度 | 约束 | 描述 |
| id | int | 11 | 非空、主键 | 自增主键 |
| url | varchar | 255 | 非空 | 电影网址 |
| user | Varchar | 50 | 非空 | 用户名 |
| time | vararch | 50 | 无 | 评论时间 |
| comment | text | 最大长度 65535 字节 | 无 | 短评内容 |
| score | varchar | 20 | 无 | 评分 |

表 4-7：电影图片表

| 表名 | t_mtimeimages | | | |
|---|---|---|---|---|
| 字段名 | 数据类型 | 长度 | 约束 | 描述 |
| id | int | 11 | 非空、主键 | 自增主键 |
| url | varchar | 255 | 非空 | 电影网址 |
| imgUrl | varchar | 255 | 非空 | 图片网址 |

## 4.6 iSearch 系统出错处理设计

### 4.6.1 出错输出信息

表 4-8：出错信息一览表

| 模块 | 错误描述 | | 输出信息 |
|---|---|---|---|
| 数据操作模块 | （1）用户指定的种子文件不存在 | | 输入文件不存在！ |
| | （2）命令调用错误 | | 命令使用提示 |
| | （3）数据库连接错误 | | 连接数据库失败！ |
| | （4）用户选择的模板文件不存在 | | 模板文件不存在或已删除！ |
| | （5）用户新建模板文件与已有模板文件重名 | | 指定模板文件名已存在！请重新输入！ |
| Web 可视化检索模块 | （6）用户提交检索的图片格式不正确 | | 不支持的文件格式！请重新提交！ |
| | （7）系统获取检索结果过程发生异常 | | 系统检索过程中发生错误！请重试！ |
| | （8）用户选择查看的图片已被删除 | | 图片不存在或已被删除！请重新选择！ |
| 手机客户端 | 错误类型及输出信息与 Web 可视化检索模块类似，不详述 | | |
| 广告注册模块 | 登录 | （9）用户名错误 | 用户名不存在！ |
| | | （10）密码错误 | 密码错误！ |
| | 注册 | （11）用户名已存在 | 该用户名已经被注册！请重新输入！ |
| | | （12）密码不一致 | 两次输入的密码不一致！请重新输入！ |
| | | （13）邮箱已存在 | 该邮箱地址已绑定账户！请重新输入！ |
| | 广告上传 | （14）用户提交图片格式不正确 | 不支持的文件格式！请重新选择！ |
| | | （15）必填项未填写完整 | 请将必填项填写完整！ |
| | | （16）填写信息格式不正确 | 输入的信息格式不正确！（如邮箱地址、相关链接地址） |
| | | （17）上传失败 | 广告上传失败！请重新提交！ |
| | 广告管理 | （18）要删除的图片已经不存在 | 图片不存在或已被删除！ |
| | | （19）要查看的图片已经不存在 | 图片不存在或已被删除！ |
| | （20）修改账户密码时两次输入不一致 | | 两次输入的密码不一致！请重新输入！ |

### 4.6.2 出错处理策略

表 4-9：出错处理策略一览表

| 错误编号 | 处理策略 |
|---|---|
| （1） | 系统输出提示，不开始抓取任务。 |
| （2） | 系统输出命令使用提示，不执行程序。 |
| （3） | 系统输出错误提示，不执行后续处理。 |
| （4） | 系统输出相应提示，不执行后续处理。 |
| （5） | 系统输出相应提示，不执行覆盖写模板文件。 |
| （6） | 系统输出相应提示，不提交检索请求。 |
| （7） | 系统提示检索发生错误，不返回检索结果。 |
| （8） | 系统提示图片已被删除，不从 iSimilar 图片仓库获取图片数据。 |
| （9） | 系统提示用户名不存在，阻止用户登录。 |
| （10） | 系统提示密码错误，阻止用户登录。 |
| （11） | 系统提示用户名已被注册，不新增账户。 |
| （12） | 系统提示两次输入密码不一致，阻止提交注册请求。 |
| （13） | 系统提示邮箱已被绑定，不新增账户。 |
| （14） | 系统提示提交的文件格式不正确，不执行后续处理。 |
| （15） | 系统提示用户将信息填写完整，阻止提交广告上传请求。 |
| （16） | 系统提示用户检查输入数据的格式，阻止提交广告上传请求。 |
| （17） | 系统提示用户上传操作失败，广告数据没有被正确保存。 |
| （18） | 系统提示用户要删除的图片已不存在，阻止提交删除请求。 |
| （19） | 系统提示用户要查看的图片已不存在，阻止提交查看广告信息请求。 |
| （20） | 系统提示用户两次输入的密码不一致，阻止提交修改账户信息请求。 |

（注：表 4-9 中的编号对应表 4-8 中的错误编号。）

# 第五章　iSearch 系统模块设计

第四章分析了 iSearch 系统的架构设计，在本章中将对 iSearch 系统的各模块的静态结构和关键类设计进行讨论。

在 5.1 小节中对 iSearch 系统的主要模块进行了回顾，并简单介绍了系统的界面设计。5.2 至 5.5 小节对各模块的静态结构和关键类进行分析和设计。在 5.2 小节中将重点介绍本论文提出的基于 XPATH 的模板信息提取方法。

## 5.1　iSearch 系统模块概述

iSearch 系统主要可分为四大模块：Web 可视化检索模块、广告注册模块、手机客户端和数据操作模块。在数据操作模块中，提出了基于 XPATH 的模板信息提取方法，实现了对网页数据准确的按需提取。Web 可视化检索模块主要向用户提供了基于内容的图像检索服务，用户可以通过系统查找相似图片，并获取图片的扩展信息。手机客户端提供与 Web 可视化检索模块一致的功能，用户可使用系统提供的手机应用程序，通过拍照或选择图片来进行检索，获取感兴趣的内容。广告注册模块包含登录、注册、账户信息管理、广告信息管理及广告上传五个主要子功能，该模块的主要工作总结如下：控制用户的单点登录；提供新建账户功能，并避免同一账户名重复注册；管理账户密码及简介的修改；提供对登录账户已上传广告信息的浏览、修改和删除操作；提供广告图片及相关扩展信息的上传操作。数据操作模块是 iSearch 系统的数据来源基础，包括分布式网络爬虫、数据分析处理部件、数据统计部件和模板管理部件；网络爬虫负责快速抓取数据并准确提取所需信息，数据分析处理部件负责对网络爬虫抓取的数据进行分析并合并相关信息，数据统计部件负责统计抓取的图片数及电影数，模板管理部件提供模板文件的新建、修改、删除和测试功能。

系统与用户之间有三类交互接口：浏览器、手机和 Linux 命令行终端，整体界面设计如图 5-1 所示。

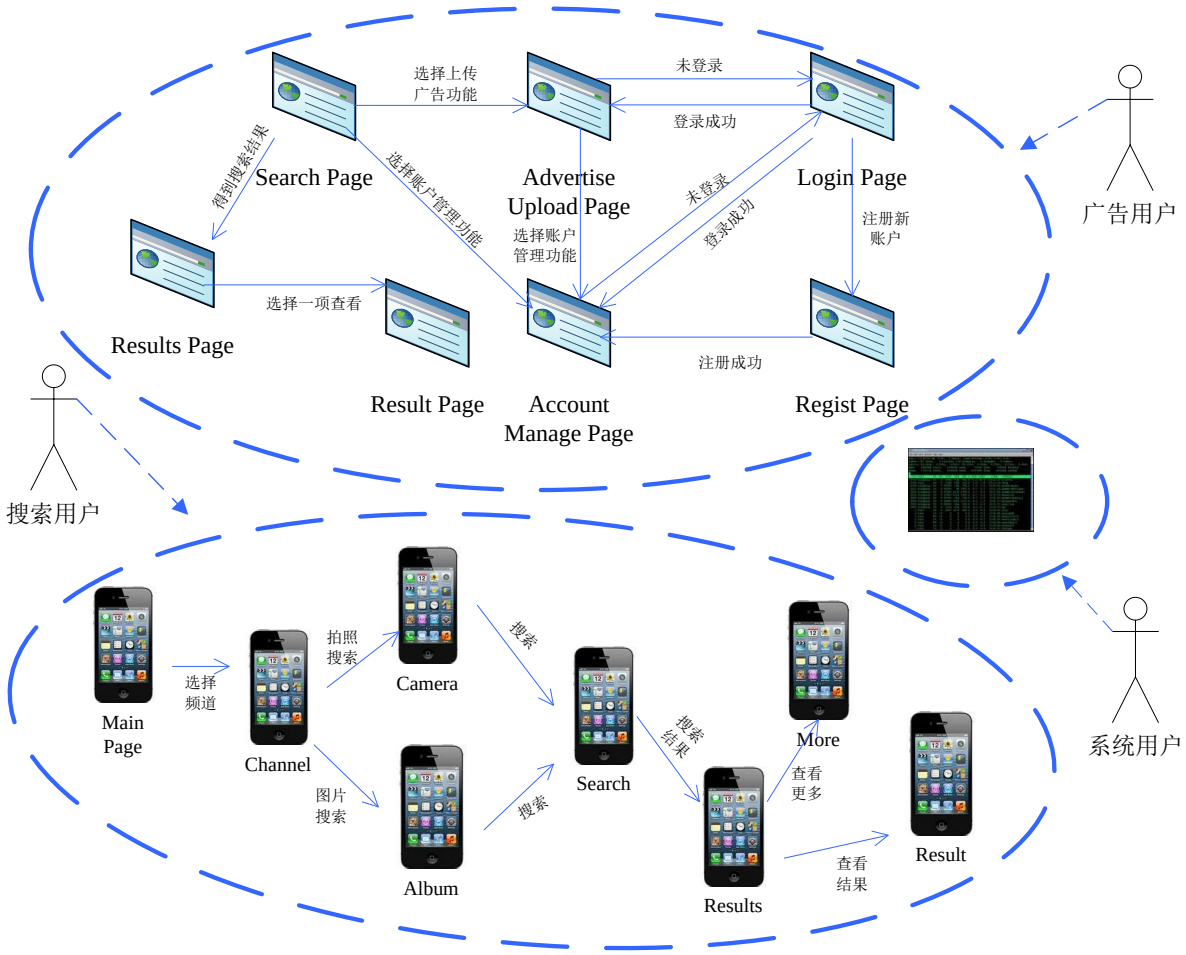



图 5-1：iSearch 系统整体界面设计图

## 5.2 数据操作模块

数据操作模块可分为如下四个子模块：网络爬虫、数据分析处理、数据统计、模板管理。数据分析处理子模块对网络爬虫子模块抓取的数据进行分析和处理；数据统计部件根据数据分析处理子模板的结果来统计图片及电影数量；模板管理子模块管理网络爬虫子模块用于提取网页数据的模板文件。这四个子模块之间相辅相成，下面分别对四个子模块进行讨论。

### 5.2.1 扩展的 Nutch 网络爬虫

本论文没有对 Nutch 网络爬虫的源代码进行改动，只是通过 Nutch 提供的扩展点实

现了基于 XPATH 的模板信息提取功能扩展插件，并通过相应的配置集成到原 Nutch 中。

下面将对基于 XPATH 的模板信息提取功能的详细设计进行讨论，并介绍 Nutch 扩展插件的配置方法。基于 XPATH 的模板信息提取方法静态结构如图 5-2。

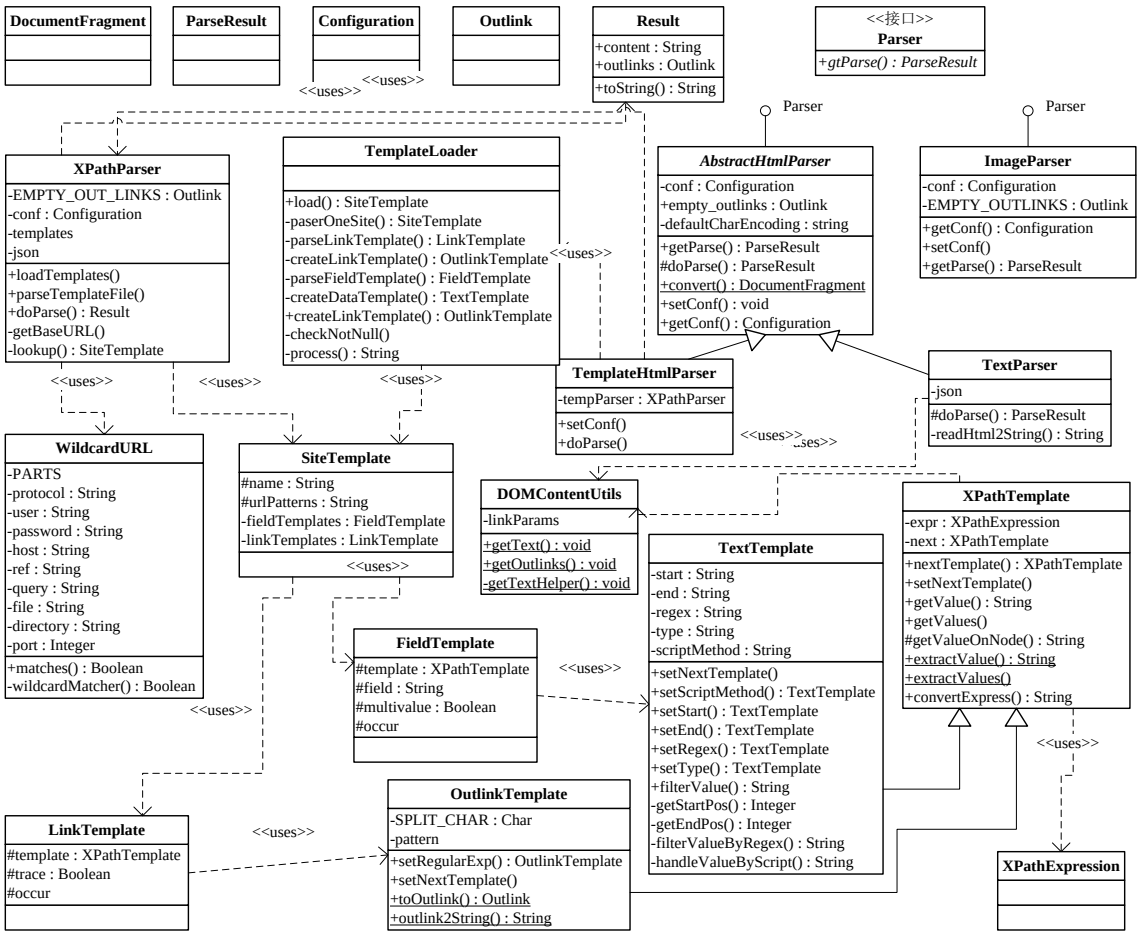



图 5-2：基于 XPATH 的模板信息提取方法静态结构设计图

ImageParser 实现了扩展点 org.apache.nutch.parse.Parser，主要负责处理从网络中下载的图像文件数据，getParse 函数将图像字节数据编码为 Base64 字符串。

AbstractHtmlParser 是实现了扩展点 org.apache.nutch.parse.Parser 的抽象类，声明了对网页文本的抽象处理方法 doParse。

TextParser 继承了 AbstractHtmlParser，实现了 doParse 抽象函数，提供了抽取网页标题和正文的功能。

TemplateHtmlParser 也继承了 AbstractHtmlParser，它主要提供的是基于模板的信息提取功能。TemplateHtmlParser 根据配置读取相应的模板文件，对模板文件进行分析并

转换为相应类型的模板对象，然后通过 XPathParser 类型的成员变量来实现对指定信息的提取。

XPathTemplate 是所有模板类型的抽象父类，具体介绍见表 5-1。

表 5-1：XPathTemplate 类介绍

| 类名 | XPathTemplate | | |
|---|---|---|---|
| 描述 | 所有模板类型的抽象父类，声明了对结点的处理方法 | | |
| 属性 | | | |
| **名字** | **类型** | **描述** | |
| expr | XPathExpression | XPATH 路径表达式的代码表示 | |
| next | XPathTemplate | 下一个模板 | |
| 方法 | | | |
| **名字** | nextTemplate | **参数** | 无 |
| **返回类型** | XPathTemplate | **描述** | 返回下一个模板实例 |
| **名字** | setNextTemplate | **参数** | 下一个模板实例 next |
| **返回类型** | void | **描述** | 设置下一个模板实例 |
| **名字** | getValue | **参数** | 要进行内容提取操作的项 item，和上下文对象 context |
| **返回类型** | String | **描述** | 提取出来的数据字符串 |
| **名字** | getValues | **参数** | 要使用 XPATH 进行分析和提取的节点 item（一般为根节点），和上下文对象 context |
| **返回类型** | String[] | **描述** | 返回从 item 节点经 expr 路径表达式选取的结点中提取出来的数据数组 |
| **名字** | getValueOnNode | **参数** | 要进行内容提取操作的节点 item，和上下文对象 context |
| **返回类型** | String | **描述** | 返回节点 item 经分析和处理后提取出来的数据，抽象方法 |
| **名字** | extractValue | **参数** | 用于提取数据的模板对象 template，要处理的节点 input 和上下文对象 context |
| **返回类型** | String | **描述** | 返回 input 使用 template 模板提取出来的第一项数据，调用了 extractValues 方法 |
| **名字** | extractValues | **参数** | 用于提取数据的模板对象 template，要处理的节点 input 和上下文对象 context |
| **返回类型** | String[] | **描述** | 返回 input 使用 template 模板提取出来的所有数据 |
| **名字** | convertExpress | **参数** | XPATH 路径表达式 expression |
| **返回类型** | String | **描述** | 返回经规范化处理的 expression，以便后续处理 |

TemplateLoader 类主要负责根据配置读取模板文件，并对内容进行分析，将模板文件中的结点转换为程序中相应的代码表示形式，具体介绍见表 5-2。

表 5-2：TemplateLoader 类介绍

| 类名 | TemplateLoader | | |
|---|---|---|---|
| 描述 | 模板文件加载器，根据配置读取所有的模板文件，并将模板转换为程序中的相应模板类型实例 | | |
| 方法 | | | |
| **名字** | load | **参数** | 输入的模板文件流 in |
| **返回类型** | SiteTemplate[] | **描述** | 对输入流 in 进行分析并根据配置创建各模板类型实例，最终返回 SiteTemplate 模板数组，每个元素代表一个网页提取模板 |
| **名字** | parseOneSite | **参数** | 代表一个网页提取模板的结点 siteNode |
| **返回类型** | SiteTemplate | **描述** | 分析 siteNode 中的 field 和 entry 结点，分别构建 FieldTemplate 和 LinkTemplate 实例，组合成 siteNode 对应的提取模板实例返回 |
| **名字** | parseLinkTemplate | **参数** | 代表一个链接提取规则的结点 node |
| **返回类型** | LinkTemplate | **描述** | 根据 node 中的 XPATH 路径表达式构建 OutlinkTemplate 实例，返回生成的 node 对应的链接提取规则实例 |
| **名字** | createLinkTemplate | **参数** | 代表一个链接提取规则的结点 node |
| **返回类型** | OutlinkTemplate | **描述** | 根据 node 中的 XPATH 路径表达式及正则匹配规则构建 OutlinkTemplate 实例并返回 |
| **名字** | parserFieldTemplate | **参数** | 代表一个文本内容提取规则的结点 node |
| **返回类型** | FieldTemplate | **描述** | 根据 node 中的 XPATH 路径表达式构建 TextTemplate 实例，返回生成的 node 对应的文本内容提取规则实例 |
| **名字** | createDataTemplate | **参数** | 代表一个文本内容提取规则的结点 node |
| **返回类型** | TextTemplate | **描述** | 根据 node 中的 XPATH 路径表达式、字符串截取范围、正则表达式及 JavaScript 操作等规则构建 TextTemplate 实例并返回 |
| **名字** | checkNotNull | **参数** | XPATH 路径表达式 s，消息字符串 msg |
| **返回类型** | void | **描述** | 若 s 为空，抛出异常消息 msg |
| **名字** | process | **参数** | 需要进行处理的字符串 input |
| **返回类型** | String | **描述** | 当 input 不为空时将 input 中的制表符、回车符、换行符删除，返回处理后的字符串 |

TextTemplate 继承了 XPathTemplate，实现了 getValueOnNode 抽象方法，提供了提取参数 Node 结点中的文本内容的功能，它是模板文件 field 结点中的 template 结点的代码表示形式。此外，filterValue 函数提供了对提取后的文本根据模板文件中定义的起始和结束字符进行内容截取的功能；而 handleValueByScript 函数则是利用 JavaScript 函数引擎，对提取结果使用 JS 函数进行进一步处理。filterValue 和 handleValueByScript 函数提供了更准确地获取网页中所需文本信息的方法。

OutlinkTemplate 同样继承了 XPathTemplate，它是模板文件 entry 结点中的 template 结点的代码表示形式。OutlinkTemplate 中的 getValueOnNode 函数提供的是提取参数 Node 结点中包含链接的功能。

FieldTemplate 包含类型为 TextTemplate 的成员变量，它是模板文件中 field 结点的代码表示形式。

LinkTemplate 包含类型为 OutlinkTemplae 的成员变量，它是模板文件中 entry 结点的代码表示形式。

SiteTemplate 包含四个成员变量：name 为模板名；urlPatterns 数组存储的是该模板的 URL 匹配规则；fieldTemplates 是类型为 FieldTemplate 的列表；linkTemplates 是类型为 LinkTemplate 的列表。SiteTemplate 是模板文件中 website 结点的代码表示形式。

WildcardURL 类提供了判断 URL 是否满足匹配条件的方法，主要用于为网页数据选择合适的信息提取模板。

XPathParser 类实现了基于 XPATH 的模板信息提取方法，具体介绍见表 5-3。

表 5-3：XPathParser 类介绍

| 类名 | XPathParser | |
|---|---|---|
| 描述 | 网页信息分析和提取部件，实现了基于 XPATH 的模板信息提取方法，根据 XPATH 路径表达式及相关的字符串处理，达到准确提取指定数据及链接的效果 | |
| 属性 | | |
| **名字** | **类型** | **描述** |
| EMPTY_OUT_LINKS | Outlinks[] | 当提取链接结果为空时用于向爬虫返回空链接数组 |
| conf | Configuration | 配置对象，包含设置的配置数据项 |
| templates | Map<WildcardURL,SiteTemplate> | 映射表，键为模板的 URL 匹配规则对象，值为相应的模板实例 |

| | | | |
|---|---|---|---|
| json | ObjectMapper | | 用户将 Map 格式的数据转换为 JSON 格式的工具 |
| 方法 | | | |
| **名字** | loadTemplates | **参数** | 无 |
| **返回类型** | void | **描述** | 根据配置获取模板文件列表，调用 parseTemplateFile 方法初始化模板列表 |
| **名字** | parseTemplateFile | **参数** | 模板加载器 loader，文件系统 fs，模板文件路径 xmlFilePath |
| **返回类型** | void | **描述** | 根据 xmlFilePath 获取文件数据流，通过 loader 构建模板并加入成员变量 templates |
| **名字** | doParse | **参数** | 网页地址 url，网页结构树 root，网页所在的根目录 linkBase |
| **返回类型** | Result | **描述** | 从 templates 中找出适合的模板实例进行处理，得到的结果封装成 Result 对象返回；若无适合的模板，返回 null |
| **名字** | getBaseURL | **参数** | 网页结构树 root，默认地址 linkBase |
| **返回类型** | URL | **描述** | 在 root 中查找 base 结点，若找到则返回 base 结点中的 href 地址，否则返回默认项 linkBase |
| **名字** | lookup | **参数** | 网页地址 url |
| **返回类型** | SiteTemplate | **描述** | 用 templates 的键对 url 进行匹配，匹配成功的返回该键对应的 SiteTemplate 模板实例；若所有键都不能成功匹配，返回 null |

下面对 Nutch 使用扩展功能的相关配置进行介绍。以本论文中的信息提取扩展插件为例，配置步骤如下：

（1） 首先需要为插件编写三个配置文件：1、build.xml，定义了可供 ant 执行的批处理命令，用于打包插件；2、ivy.xml，定义了插件所需的第三方包列表；3、plugin.xml，在这个文件里声明了插件所实现的扩展点为 org.apache.nutch.parse.Parser，及插件中实现了该扩展点的具体类。

（2）将编写好的插件放在网络爬虫 nutch 目录下的 src/plugin 文件夹内。

（3）修改 Nutch 的相应配置：1、在 src/plugin 文件夹下的 build.xml 文件中加入对插件的打包处理。2、在 conf 文件夹下的 parse-plugins.xml 文件配置网页文件类型及插件中相应的分析处理类。

### 5.2.2 数据分析与合并部件

在本文的第一章中已经提到，根据网络爬虫提取数据的不同，数据分析与合并方法的实现也需要进行相应的调整，但主要的处理流程是不变的，即：根据数据中的关联标记找出相关数据并进行合并，重复执行该过程直至完成所有相关数据的合并。该部件采用了 MapReduce 编程框架。下面以分析和处理“时光网”数据的部件为例进行介绍。

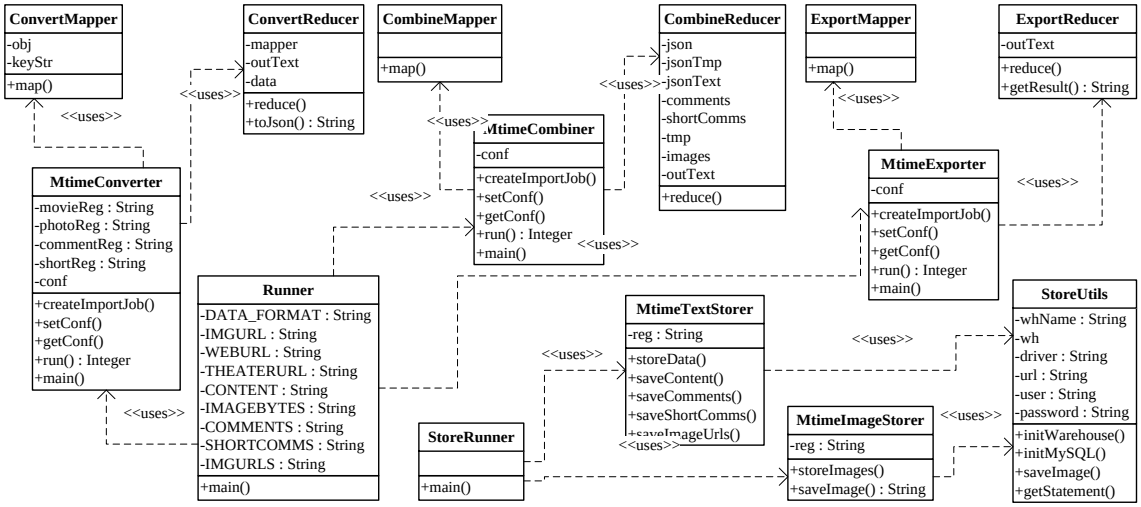



图 5-3：时光网数据分析处理部件静态结构图

| 电影详细信息 | 电影影评信息 | 电影短评信息 | 电影图片信息 | 图片数据 |
|---|---|---|---|---|
| Key: movieUrl | Key: webUrl | Key: webUrl | Key: webUrl | Key: imageUrl |
| Value:<br>Name,<br>director, …. | Value:<br>movieUrl,<br>users, …. | Value:<br>movieUrl,<br>users, …. | Value:<br>movieUrl,<br>imageUrl | Value:<br>图片字符串 |

图 5-4：爬虫抓取的时光网数据格式

通过网络爬虫 Nutch 抓取的时光网数据如图 5-4 所示。时光网的每一部电影的数据分为五个部分，其中 “电影影评信息”、“电影短评信息”、“电影图片信息”可以通过 movieUrl 与“电影详细信息”关联；而“图片数据”则需要通过“电影图片信息”间接与“电影详细信息”关联。

MtimeConverter 负责将“电影影评信息”、“电影短评信息”和“电影图片信息”中

的 Key 的值替换成 movieUrl。ConvertMapper 负责过滤无用的数据，并把过滤后的数据发送给 ConvertReducer 进行处理。ConvertReducer 负责执行 Key 值的替换。MtimeConverter 处理后的数据如图 5-5 所示。

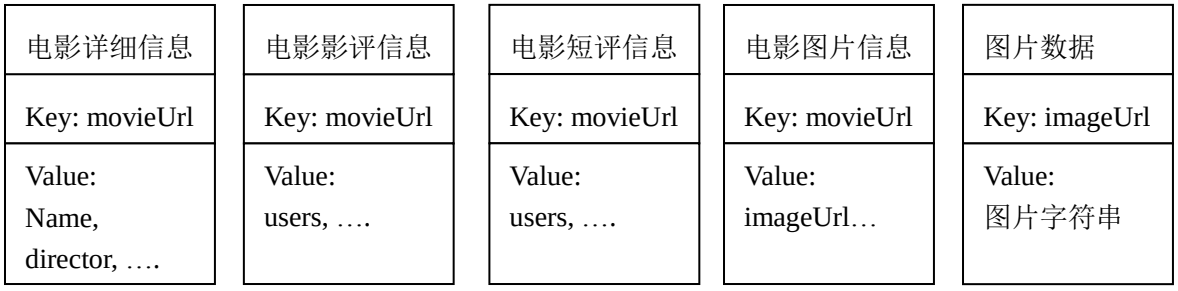



图 5-5：经 MtimeConverter 处理后的时光网数据格式

MtimeCombiner 负责将“电影影评信息”、“电影短评信息”、“电影图片信息”和“电影详细信息”合并，并将以 imageUrl 作为合并后的数据的 Key。CombineMapper 将数据发送给 CombineReducer 进行处理，在数据发送到 CombineReducer 之前会根据 Key 对数据进行分组。CombineReducer 对分组后的数据进行合并，并从中提取出 imageUrl 作为合并后数据的 Key 输出。MtimeCombiner 处理后的数据如图 5-6 所示。

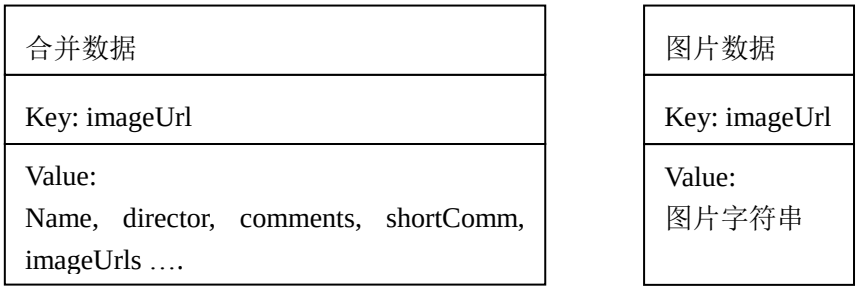



图 5-6：经 MtimeCombiner 处理后的时光网数据格式

MtimeExporter 负责将合并数据和图片数据合并。ExportMapper 将数据发送给 ExportReducer 进行处理，在数据发送到 ExportReducer 之前会根据 Key 对数据进行分组。ExportReducer 对分组后的数据进行合并。经过这三步处理后即可将原来分散为五部分的同属一部电影的数据组合在一起。

MtimeImageStorer 负责把图片数据存储到 iSimilar 平台的图片仓库中。而 MtimeTextStorer 则负责对组合后的电影的文本数据进行分析并存储到 MySQL 数据库。

### 5.2.3 数据统计部件

数据统计部件以数据分析处理部件的处理结果为输出，对图片数量和电影数量进行统计。针对时光网数据的统计部件静态结构设计如图 5-7。MtimeImageStatis 负责统计抓取的时光网海报图片总量；StatisMapper 对数据添加记数标记后发送给 StatisReducer 进行处理；StatisReducer 接收数据并进行统计，输出为图片总数量。MtimeMovieStatis 负责统计抓取的时光网电影总量；MovieStatisMapper 负责提取电影网址作为 Key 发送给 MovieStatisReducer 进行处理；MovieStatisReducer 统计数据分组数，输出为电影总数量。

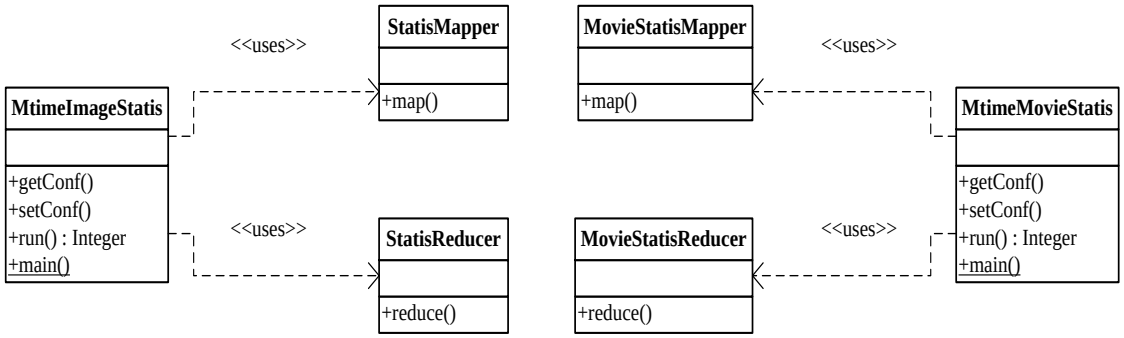



图 5-7：时光网数据统计部件静态结构图

### 5.2.4 模板管理部件

模板管理部件主要提供对用于提取网页信息的模板文件的新增、修改、删除和测试的功能，静态结构如图 5-8。InitServlet 负责部件常用变量的初始化，并提供这些变量的获取途径。GetTemplateServlet 负责根据用户请求获取相应模板文件并返回内容字符串。DeleteTemplateServlet 负责处理用户的删除模板文件请求，并给出相应的响应。SaveTemplateServlet 负责处理用户的保存模板文件请求，包括新建模板文件和修改模板文件的保存。GetFileServlet 负责根据用户请求获取相应的用于测试模板文件的网页文件内容，并处理后返回网页内容字符串。TestTemplateServlet 负责处理用户的测试模板文件请求，根据用户之前选择的模板文件及提交的测试网页文件，调用网页信息提取方法，并向用户返回处理结果。Const 提供了获取网页编码及处理网页内容字符串的函数，主要负责将网页文本转换为系统页面可以正常显示的格式。

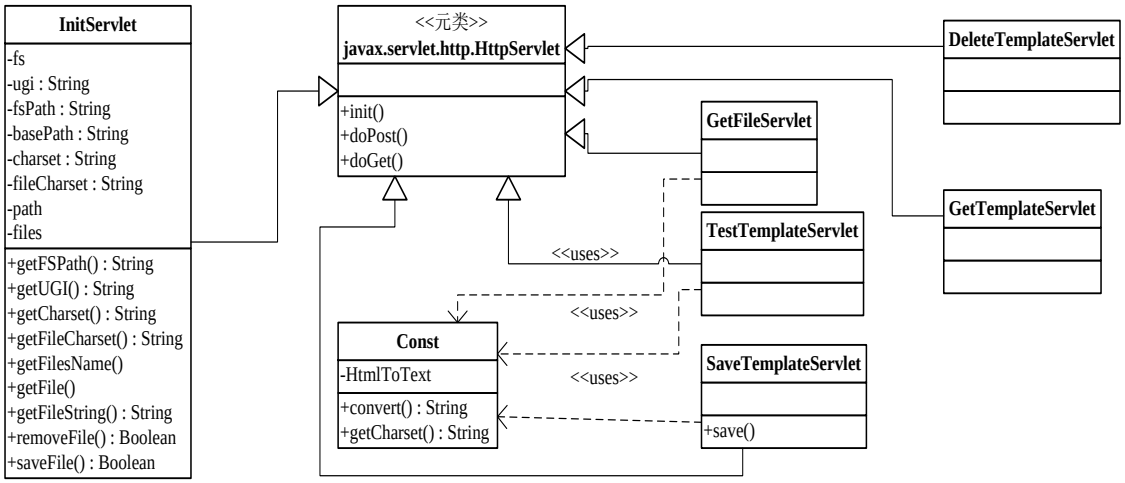



图 5-8：模板管理部件静态结构图

## 5.3 Web 可视化检索模块设计

Web 可视化检索模块采用了 SSH 框架进行设计，主要提供了基于内容的图像检索功能，静态结构如图 5-9。其中 BaseSessionAction、UploadUtilAction 和 SearchAction 为控制层，SearchService 和 SearchServiceImpl 为业务逻辑层。

InitListener 在程序初始化的时候执行，主要负责根据配置文件初始化常用变量，并提供常用变量的获取方法。BaseSessionAction 为本模块各 Action 类的抽象父类，实现了 SessionAware 接口，以支持对 session 的处理。UploadUtilAction 负责获取用户提交的图片，并返回处理结果。SearchAction 负责接收用户检索请求，并分配给业务处理部件 SearchService 处理。SearchServiceImpl 实现了 SearchService 接口，主要负责处理由 SearchAction 分配的检索任务。Utils 是工具类，提供随机数的生成、用户提交图片文件的临时存储等功能。

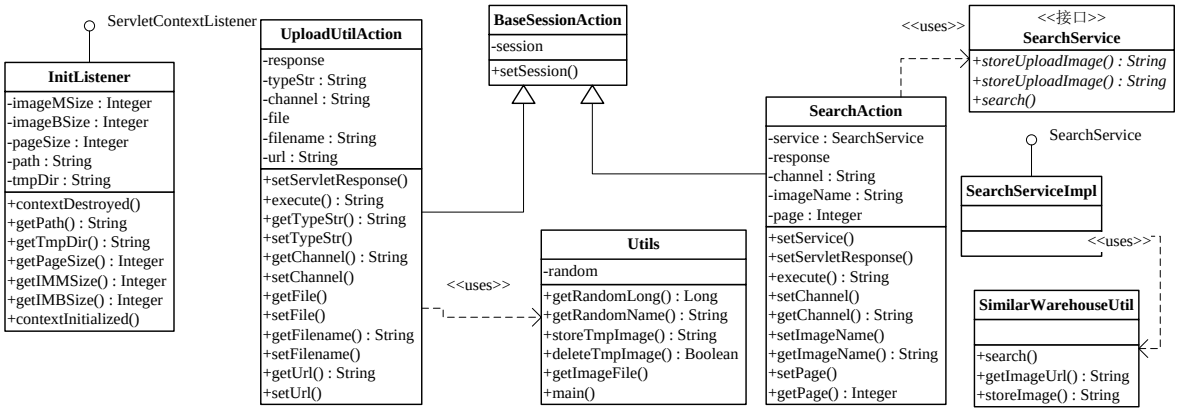



图 5-9：Web 可视化检索模块静态结构图

## 5.4 广告注册模块

广告注册模块也采用了 SSH 框架进行设计，主要提供了用户登录、注册、账户信息管理、广告信息管理及广告上传五个功能，静态结构如图 5-10。其中 InitListener 主要负责在程序启动的时候根据配置文件初始化常用变量。SingleListener 负责控制用户的单点登录。Action 类属于控制层；Service 接口及相应的 ServiceImpl 实现类属于业务逻辑层；DAO 类及数据表映射类属于模型层。下面根据功能划分进行介绍。

（1） 用户登录

LoginAction 负责接收用户登录请求并分配给 LoginService 进行处理。LoginServiceImpl 实现了 LoginService 接口，主要通过 TUserDAO 的数据处理结果判断用户是否登录成功，并返回相应提示。TUserDAO 提供用户密码正确性的判断方法。

（2） 用户注册

RegistAction 负责接收用户的注册请求并分配给 RegistService 进行处理。RegistServiceImpl 实现了 RegistService 接口，主要通过 TUserDAO 的数据处理结果来判断账户名是否已存在、邮箱地址是否已被使用，若满足注册条件，则通过 TUserDAO 新增账户，并返回相应提示。TUserDAO 提供了账户名及邮箱地址是否已存在的判定方法，及插入新账户信息的方法。

（3） 账户信息管理

AccountManageAction 负责接收用户的查看账户信息及修改账户信息请求，并分配给 AccountManageService 进行处理。AccountManageServiceImpl 实现了

AccountManageService 接口，它通过 TUserDAO 来进行相应的数据处理，并返回相应提示。TUserDAO 提供了账户信息的获取方法，及更新账户信息的方法。

（4） 广告信息管理和广告上传

AdManageAction 负责接收用户对广告数据进行操作请求，包括获取已上传广告列表、查看广告信息、修改广告信息、删除广告信息和上传广告信息，并将请求分配给 AdManageService 进行处理。AdManageServiceImpl 实现了 AdManageService 接口，它对根据请求操作的不同，分别调用 TAdDAO 和 TDeleteDAO 的函数来进行相应的数据处理。在处理上传广告操作时还需要调用 iSimilar 的数据存储接口对广告图片数据进行存储；在处理删除广告信息操作时，通过 TAdDAO 将相应的广告信息从数据表 t_ad 中删除，并在数据表 t_delete 中插入删除广告的图片 ID。TAdDAO 提供了广告信息的获取、插入、更新和删除的操作方法；TDeleteDAO 提供了获取和新增被删除的广告图片 ID 的方法。

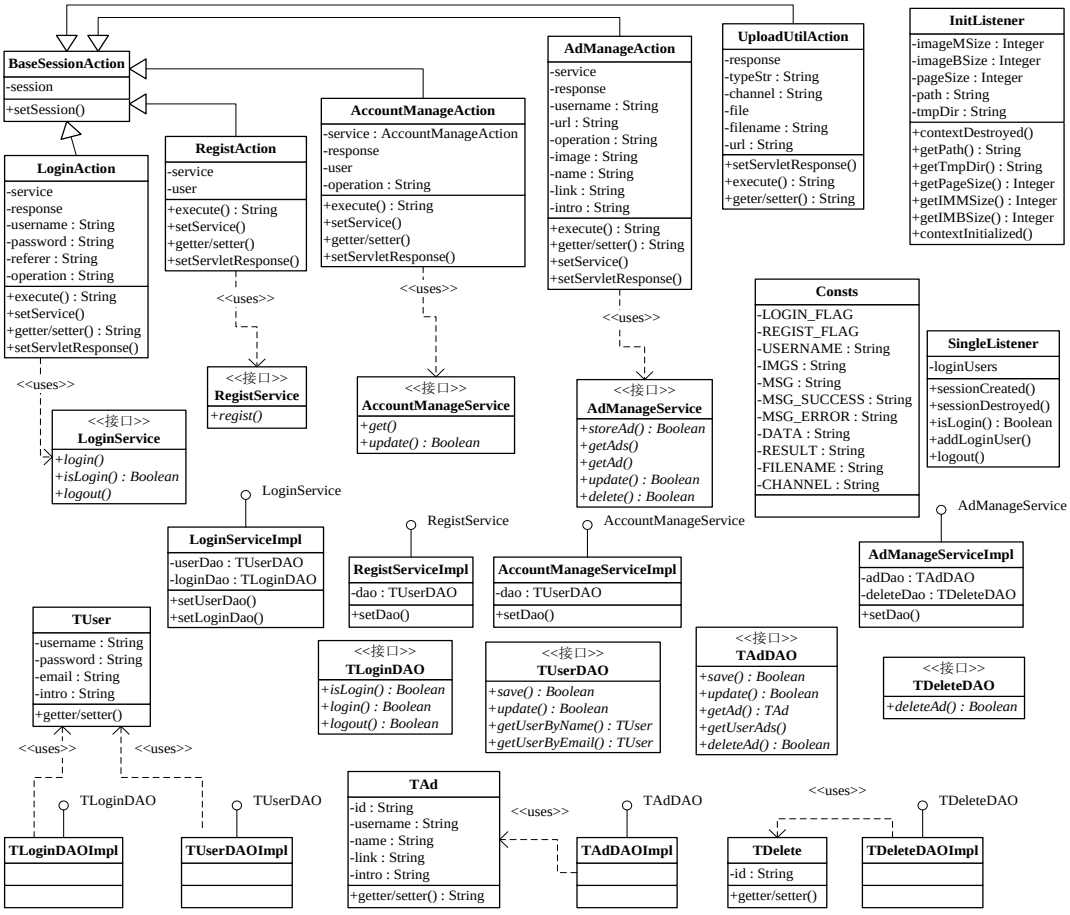



图 5-10：广告注册模块静态结构图

## 5.5 手机客户端

Android 及 Windows Phone 客户端均是采用一个页面对应一个处理部件的方式进行设计。下面对 Android 客户端的主要处理部件进行介绍，Windows Phone 客户端的设计与 Android 类似，本文不再进行说明。

手机客户端的使用方法如下：1、打开程序首页；2、选择频道（电影、衣服）；3、通过拍照或从手机相册中选择一张图片；4、提交搜索请求；5、客户端获取检索结果并显示前四个结果；5、点击“查看更多”按钮来浏览更多检索结果；6、选择一项查看详细信息；7、若检索结果是电影，还可查看影评、微影评、海报和影院信息；若检索结果是衣服，可查看购买网页。

对使用方法分析后可知，手机客户端主要有 11 个页面，分别对应一个 Activity 处理部件：

ISearchActivity 是首页操作的处理部件，提供频道的选择及退出程序功能。

ChannelActivity 是频道页面操作的处理部件，提供拍照或从手机相册选择检索图片，及返回首页的功能。

SearchActivity 是检索页面操作的处理部件，提供提交检索请求，重新拍摄或重新选择图片，及返回频道的功能。

ResultsActivity 是检索结果页面操作的处理部件，提供查看详情，查看更多结果，返回频道和返回首页的功能。

MoreActivity 是查看更多结果页面操作的处理部件，提供查看详情，返回频道，返回首页和返回上一页的功能。

MovieViewActivity 是查看电影详情页面操作的处理部件，提供查看影评，查看短评，查看海报，查看影院信息，返回频道，返回首页和返回上一页的功能。

ClothViewActivity 是查看衣服详情页面操作的处理部件，提供跳转到衣服购买网页，返回频道，返回首页及返回上一页的功能。

# 第六章　iSearch 系统部署与应用

第三至第五章完成了对 iSearch 系统的分析与设计。在这章中将对系统的部署及应用进行介绍。6.1 小节介绍了系统的开发环境及运行环境的软、硬件配置。6.2 小节对系统的功能进行了测试，包括对本论文提出的基于 XPATH 的模板信息提取方法的效果、系统的检索功能、广告上传功能等方面的测试。

## 6.1　开发环境与运行环境

表 6-1：系统开发环境配置

| **软件环境** | |
|---|---|
| 操作系统 | Microsoft Window 7、Ubuntu 10.10 |
| 开发平台 | Eclipse Juno 4.2 Edition、Visual Studio 2010 Express for Windows Phone |
| JavaScript 类库 | prototype、jquery1.7.2、jquery-form-1.7.2 |
| 语言库 | JDK1.6 |
| WEB 服务器 | Tomcat7.0.29 |
| 数据库 | MySQL Server 5.0 |
| 浏览器 | Mozilla Firefox15.0 |
| 分布式集群 | Hadoop1.0.3、hbase0.94.1 |
| SSH 通信工具 | SecureCRT |
| **硬件环境** | |
| 个人计算机配置 | CPU：Intel Core i5-2410M 2.30GHz；内存：4G |
| 分布式集群配置 | DELL R710 服务器 10 台；每台 2 个 4 核 CPU，主频 2.3GHz，内存 20G，硬盘 4T |

表 6-2：系统运行环境配置

| 数据操作模块 | |
|---|---|
| **软件环境** | |
| 操作系统 | CentOS6.5（所有支持 hadoop1.0.3 及 hbase0.94.1 的系统） |
| 数据库 | MySQL5.0 以上版本 |
| 语言库 | JDK1.6 以上版本 |
| 集群配置 | Hadoop1.0.3、hbase0.94.1 |
| **硬件环境** | |
| CPU | 1.6GHz 以上 |
| 内存 | 总内存达到 80G 以上 |
| **WEB 服务器端** | |
| **软件环境** | |
| 操作系统 | 所有支持 tomcat7.0.29 的操作系统 |
| WEB 服务器 | Tomcat7.0.29 |
| 语言库 | JDK1.6 以上版本 |
| 读写权限 | 向该应用开放 |
| **硬件环境** | |
| CPU | 1.6GHz 以上 |
| 内存 | 1G 以上 |
| **客户端** | |
| **软件环境** | |
| 操作系统 | 无要求 |
| 浏览器 | Mozilla FireFox，Chrome |
| **硬件环境** | |
| CPU | 无要求 |
| 内存 | 无要求 |
| **手机客户端** | |
| **软件环境** | |
| 操作系统 | Android2.3.3 以上，Windows Phone7 |
| **硬件环境** | |
| 分辨率 | Android 客户端支持 320*480 分辨率，对其它分辨率支持性不是很好 |
| 机身内存 | 512M 以上 |
| 相机 | 支持客户端拍照 |

## 6.2 iSearch 系统测试

### 6.2.1 基于 XPATH 的模板信息提取方法测试

模板管理模块中的模板文件测试功能使用了本论文提出的基于 XPATH 的模板信息提取方法，可通过对模块管理模块测试模板功能的测试来验证基于 XPATH 的模板信息提取方法。测试过程如下：

1、从模块文件列表中选择要用于测试的模板文件，点击“测试”按钮。本次测试选用适用于时光网电影信息提取的模板文件。模板测试界面如图 6-1。


图 6-1：模板文件测试界面

2、选择从 url 获取测试网页，输入测试网页的 URL，点击“测试”，结果如图 6-2 所示。


图 6-2：测试模板文件功能测试结果图

结果分析：

图 6-3 展示了“http://movie.mtime.com/12231/”对应的网页及其源代码。图中左下的蓝色部分是通过图 6-1 所示模板的第一个 field 结点中定义的 XPATH 路径表达式找到的相应结点在源代码中的位置；上半部分页面中的蓝色覆盖部分则是结点在页面上的显示形式。可以清楚地看到，图 6-2 中的测试结果里，已经准确地提取出电影名“肖申克的救赎”。其他元素的对比提取不再进行详述。


图 6-3：图 6-1 所示模板第一个 field 结点对应的网页结点

## 6.2.2 模板管理模块测试


图 6-4：模板管理主界面

**（1）修改模板功能测试**

测试步骤：

1、点击 test.xml 项对应的 Edit 按钮，提交修改请求。

2、在编辑框中对模板进行修改，点击 Save 按钮提交保存请求。

3、再查看 test.xml 文件，验证修改是否被正确保存。

结果分析：

图 6-5(a)和(b)的对比验证了修改模板功能正常。


(a)


(b)

图 6-5：test.xml 模板文件修改前后的内容对比

**（2）删除模板功能测试**

测试步骤：

1、点击 test.xml 模板文件对应的 Delete 按钮。

2、查看模板文件列表，验证文件是否已被删除。

结果分析：

图 6-6 与图 6-4 的对比验证了删除模板文件功能正常。


(a)


(b)

图 6-6：(a)为系统对删除操作的响应；(b)为点击(a)中确定按钮后刷新的模板文件列表

## 6.2.3 广告上传功能测试

**（1）广告上传功能测试**

测试步骤：

1、从本地选择一张图片，点击“上传”按钮。

2、根据提示填写相应项，点击“提交”按钮。

3、打开“账户管理”中的广告管理，查看已上传广告中是否有刚刚上传的广告数据。

结果分析：

图 6-7 和图 6-8 的对比验证了广告上传功能正常。


图 6-7：未执行上传广告操作前的已上传广告列表

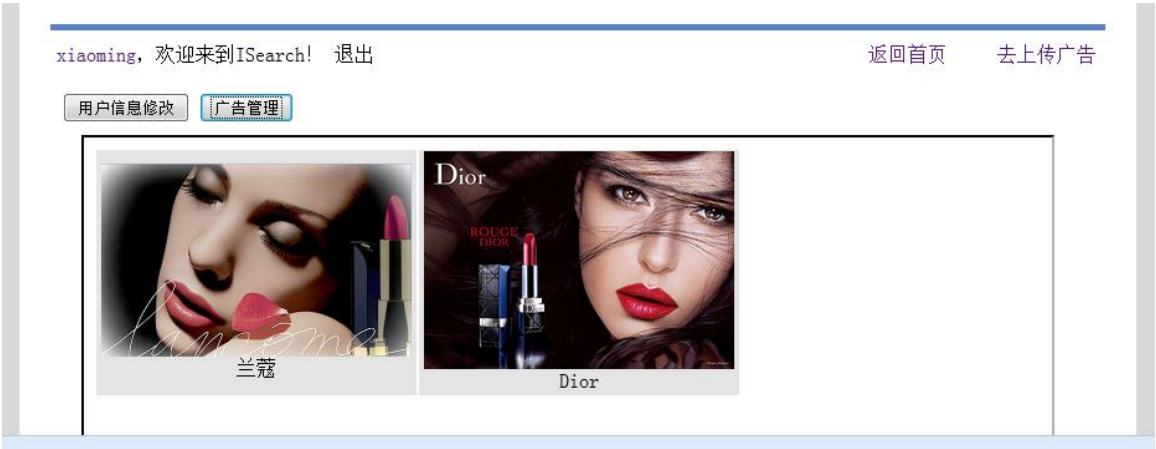



图 6-8：执行上传广告操作后的已上传广告列表

**（2）增量索引功能测试**

在广告上传功能测试时，向系统插入了一条新的图片数据，按照系统的设计，插入新数据后会自动对新数据构建增量索引。下面通过利用这张图片进行检索来测试增量索

引功能。

测试步骤：

1、选择广告检索频道，选择广告上传功能测试中提交的图片，提交检索请求。

2、查看检索结果列表，验证上传的图片是否已经能被检索。

结果分析：

图 6-9 和图 6-10 的对比验证了增量索引正常。


图 6-9：未执行上传广告操作时的检索结果


图 6-10：执行上传广告操作后的检索结果

### 6.2.4 广告管理功能测试

**（1）广告信息修改功能测试**

测试步骤：

1、从已上传的广告列表中选择一项进行浏览。

2、修改完成后点击“提交”按钮。

3、返回已上传广告页面，从列表中找到修改的广告，点击查看，验证修改信息是否被正确保存。

结果分析：

图 6-11 和图 6-12 的对比验证了广告信息修改功能正常。


图 6-11：修改前的广告信息


图 6-12：修改后的广告信息

**（2）删除广告功能测试**

测试步骤：

1、从已上传的广告列表中选择一项进行浏览，点击“删除”按钮。

2、返回已上传广告页面并刷新，验证广告是否已被删除。

结果分析：

图 6-13 和图 6-14 的对比验证了广告删除功能正常。

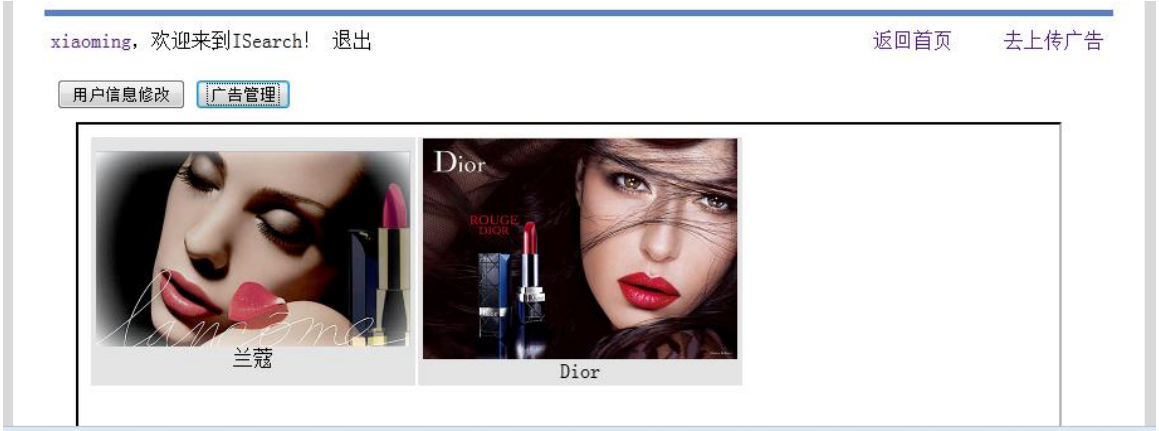



图 6-13：执行删除操作前的已上传广告列表

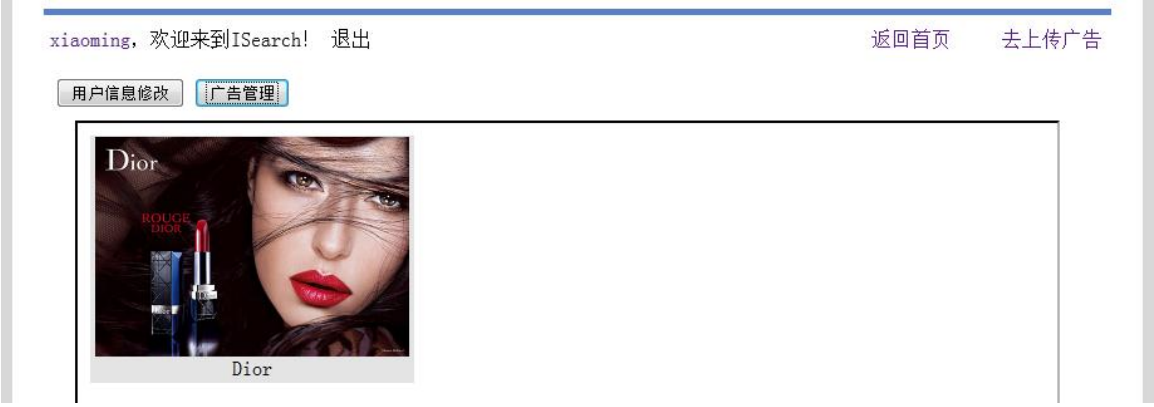



图 6-14：执行删除操作后的已上传广告列表

## 6.2.5 图像检索功能测试

**（1）Web 端图像检索功能测试**

对衣服频道进行测试，检索图片来自网络，测试步骤如下：

1、选择衣服频道，选择从网页获取检索图片。

2、将要检索的图片的网址输入搜索框，点击搜索按钮。

3、查看检索结果列表。

结果分析：

从图 6-15 中可以看到系统的 Web 端图像检索功能正常。

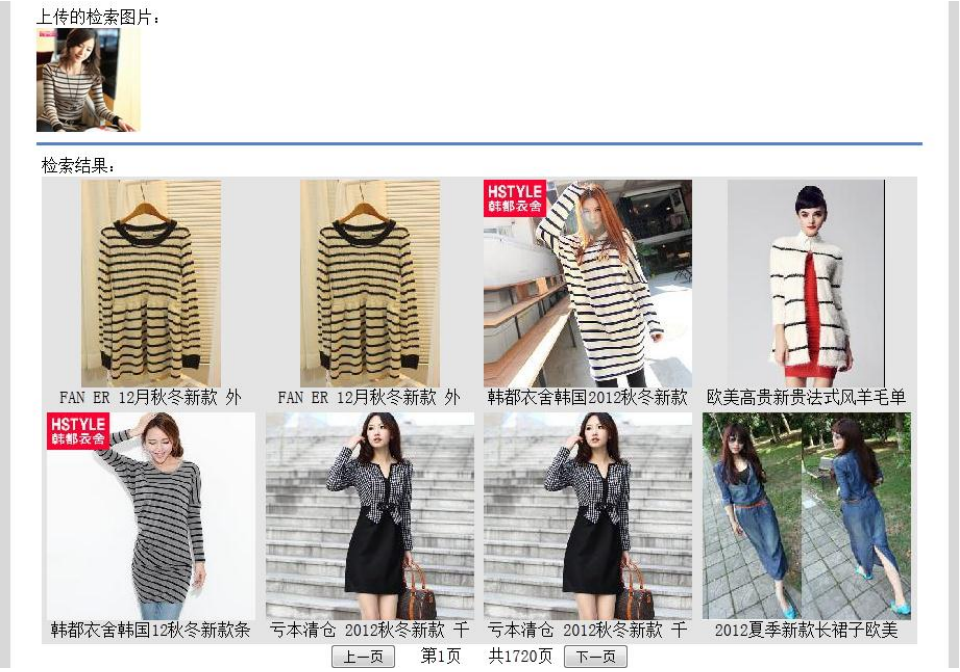
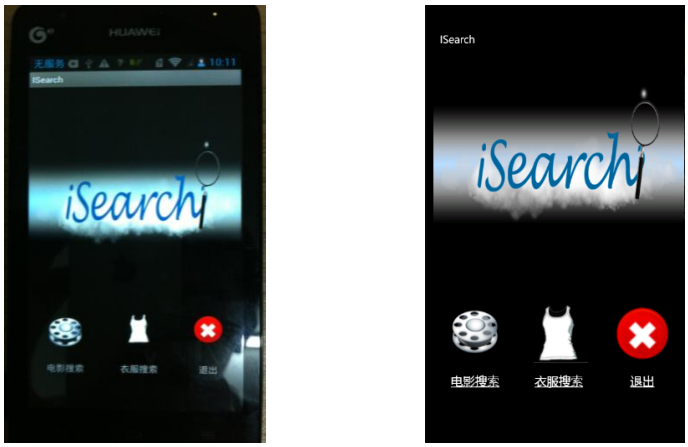



图 6-15：Web 端图片检索功能测试

**（2）手机客户端图像检索功能测试**

Android 客户端和 Windows Phone 客户端的首页页面如图 6-16。




(a)Android 客户端首页　　(b)Windows Phone 客户端首页

图 6-16：手机客户端首页

下面在 Android 客户端上进行测试，通过拍照获取检索图片，对电影频道进行测试，步骤如下：

1、打开客户端，点击“电影搜索”。

2、点击“拍照搜索”，拍摄要检索的图像，点击“搜索”。

3、浏览检索结果，选择一项查看详细信息。

结果分析：

图 6-17 到图 6-20 展示测试过程，手机客户端的图像检索功能均可正常使用。


图 6-17：电影搜索频道页面


图 6-18：拍照搜索页面


图 6-19：检索结果


图 6-20：查看结果

### （3）广告频道与百度识图对比测试

下面以一张家乐福超市 2013 年 5 月的促销海报为搜索图像，分别在“百度识图”和 iSearch 系统中进行搜索。图 6-21 为“百度识图”的检索结果，可以看到检索结果的图像与原搜索图像不符，且得到的信息明显已经失效。而图 6-22 为 iSearch 系统的检索结果，可以看到检索结果中含有与原搜索图像相符的信息。

造成“百度识图”和 iSearch 系统检索结果差异最重要的原因是数据源的差异，由于“百度识图”的数据源是通过网络爬虫抓取的，有延时性和数据不完整性的问题；而 iSearch 系统的广告数据是由广告发布商手动上传的，是有目的向前端用户发布信息，因此 iSearch 系统中的数据源更具有针对性和时效性。


图 6-21：“百度识图”检索结果


图 6-22：iSearch 系统检索结果

图 6-23 为 iSearch 系统提供的广告增强信息，图 6-24 则是图 6-23 中相关链接所对应的网页。可以看出，iSearch 系统可以为前端用户方便快捷地提供准确的广告信息，能够满足广告发布商增强广告效果的需求。


图 6-23：iSearch 系统广告增强信息


图 6-24：家乐福超市促销信息

## 6.2.6 账户管理功能测试

### （1）登录功能测试

测试步骤：

1、输入不存在的用户名，验证系统能正确处理。

2、输入错误的密码，验证系统能正确处理。

3、输入正确的用户名密码，验证系统的登录功能正常。

4、在另一浏览器上使用相同的用户名密码进行登录，验证系统的单点登录功能正常。

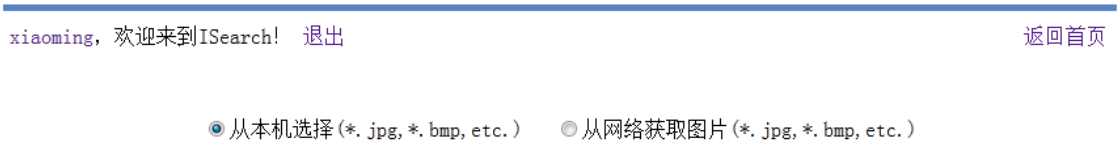
结果分析：

从图 6-25 中可以看出，系统可以正确处理用户名不存在、密码错误的情况，并阻止用户登录；并且验证了系统的单点登录功能可以正常使用。


(a)用户名不存在


(b)密码错误


(c)账号已经在别处登录或未正常退出




(d)正常登录

图 6-25：登录功能测试

**（2）注册功能测试**

测试步骤：

1、使用格式错误的邮箱地址进行注册，验证系统能正确处理。

2、两次输入密码不一致，验证系统能正确处理。

3、使用已存在的账户名进行注册，验证系统能正确处理。

4、使用已存在的邮箱地址进行注册，验证系统能正确处理。

5、使用正常的数据进行注册，验证系统的注册功能正常。

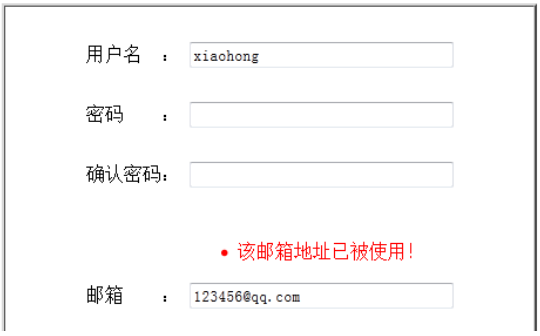
结果分析：

从图 6-26 中可以看出，系统可以正确处理账户名已存在、邮箱地址已被使用、邮箱地址格式错误、密码不一致的情况，并阻止用户注册；并且验证了系统的注册功能可以正常使用。


(a)邮箱格式不正确


(b)密码不一致


(c)用户名已被注册




(d)邮箱地址已被使用


(e)正确注册并自动登录

图 6-26：注册功能测试

**（3）账户信息修改功能测试**

测试步骤：

1、打开账户管理页面，修改密码并提交修改请求。

2、退出系统，使用原来的密码登录，验证密码是否已修改。

3、使用修改后的密码登录，验证是否能正常登录系统。

结果分析：

从图 6-27 和图 6-28 可以看出，账户信息修改功能正常。


图 6-27：修改密码后使用原密码无法登录系统


图 6-28：使用修改后的密码正常登录系统

# 第七章　总结与展望

## 总结

本论文主要来源于微软亚洲研究院与中山大学智能信息处理与云计算实验室的合作项目：广告增强系统工程。iSearch 系统是本论文为该项目提出的解决方案，旨在为提高海报广告的交互能力、增强广告效果提供新的方法。作者独立完成了 iSearch 系统的需求分析、系统分析与设计，并实现了 iSearch 系统原型。目前，iSearch 系统能够提供对衣服、电影和广告三种类别的可视化检索服务，并向广告发布商提供广告注册功能，但目前系统存在着不少有待解决的问题，离理想效果还有较大差距。

本论文的主要工作总结如下：

（1） 对开源网络爬虫 Heritrix 和 Nutch 进行了调研和实验分析，验证了 Nutch 更适合于 iSearch 对数据抓取的要求。

（2） 结合网络爬虫、分布式计算、手机应用开发等技术，设计并实现了一个基于可视化检索的在线广告信息增强系统，包括 Web 可视化检索模块、广告注册模块、数据操作模块和 Android 及 Windows Phone 两个平台的手机客户端。该系统为提高海报广告的交互性，增强广告效果提供了新的方法。

（3） 由于 iSearch 系统需要大量图片及标注数据，本论文提出了基于 XPATH 技术的可定制的模板提取方法。该方法利用 XPATH 路径表达式，准确定位到所需数据所在结点位置，再根据配置对提取后的文本做进一步处理，实现了对指定数据的准确提取。由于该方法是针对具有相同结构网页设计的，能够准确地提取大部分相似网页的数据。

（4） 由于同一事物的信息可能会分布在多个网页中，对这些网页的数据提取操作也是独立的，因此还需要对提取出来的数据进行分析和合并。本论文通过在提取数据中添加关联标记，然后逐层关联合并的方法实现了将分散在多个页面中的同一事物的描述信息合并为一条完整描述信息。

（5） 利用 MySQL 数据库存储标注信息，解决了 iSimilar 图片仓库不适宜存储过长文本的问题。并封装了相应的 HTTP 检索服务接口，以适应该存储方案，各终端可以方便地通过该接口获取 iSearch 系统检索服务。

（6） 利用 Lucene 实现增量索引，避免对图片仓库所有图片进行索引重构。

（7） 利用 MySQL 数据库记录删除图片的 ID，实现图片数据的“伪删除”，一定程度上弥补了 iSimilar 图片仓库不支持数据删除的缺陷。

（8） 结合 JSP、CSS 和 JS 进行 WEB 界面设计，并使用 Ajax 技术加快系统响应，改善用户使用体验。

（9） 对系统进行了整合和测试，验证了系统的可用性。

**展望**

本论文所实现的 iSearch 广告信息增强系统的最终目的是实现一个能有效增强海报广告效果的创新应用。该应用结合移动通信、图像检索等多项技术，向后端广告发布商提供广告注册功能，并向前端用户，特别是移动设备用户提供有准确标注信息的基于内容的实时图像检索服务。

下面对系统存在的不足进行分析并分别提出改进方案：

（1） 系统电影数据与时光网不同步

iSearch 系统电影检索频道的数据来源于“时光网”，通过网络爬虫技术来获取。从网络爬虫抓取数据，到对数据进行分析和合并，然后将图片和文本数据分别写入 iSimilar 图片仓库和 MySQL 数据库，并对新增数据建立索引，使其能被检索，这一过程需要耗费不少的时间。此外，当时光网更新数据时，网络爬虫并不能自发地启动任务对新数据进行抓取。受到这些因素的影响，当前 iSearch 系统的电影数据与时光网的数据并不能达到同步。

对于这一情况，有两种可行地改进方案：1、定时启动爬虫的抓取任务进行数据更新，这种方法可以实现系统数据与时光网数据延时性的同步；但受到网络环境等因素的影响，任务启动的间隔时间难以设定：若两个任务的启动间隔过小，会造成数据的重复抓取，并给后续处理带来困难；若任务间隔时间过长，会造成系统的数据过于滞后。2、对时光网首页进行监控，发现首页更新时即启动抓取任务；这种方法在首页更新间隔小于一次抓取任务完成时间的时候，也会出现前一种方法中提到的数据重复抓取的问题。利用网络爬虫技术获取数据会不可避免地出现数据不同步的问题，要实现数据同步的根本解决方法是从源头获取数据，这需要时光网向开发者开放如同微博 API 的数据访问接口。

（2） 系统当前的数据量不足

前面提到系统的电影频道数据存在不同步的问题，而衣服频道和广告频道则是有着缺乏充足数据的问题。目前系统的衣服频道拥有约六万张图片，远远不能满足衣物类的搜索需求；衣物类的图片及标注数据可通过网络爬虫技术进行抓取。而广告频道的数据来源并非是网络爬虫等自动化数据获取工具，而是系统的广告注册用户，只有当广告注册用户手动向系统提交广告数据时，系统的广告频道才能真正地发挥期望的作用。

（3） 增量索引方法的不足

目前系统利用 Lucene 改进了 iSimilar 原有的索引方法，采用对新插入的一个数据项新建一个索引文件，当文件数达到指定值时进行合并的方式来避免对所有数据进行重构索引的操作。索引文件数量的增加使得文件读写次数增加，并增加了一定的文件管理难度。

对此，本论文考虑用内存索引的方法来进行改进：在内存中开辟一个索引缓冲区，将新插入图片的索引数据写入缓冲区，当缓冲区满或索引项数量达到指定的数值时，将缓冲区的数据写入磁盘文件。对于系统突然关机造成内存数据丢失的情况，可加入定时将内存数据写入磁盘的方法来尽可能避免数据丢失。

（4） 删除

现在系统利用 MySQL 记录删除图片的 ID，并将删除的图片 ID 从检索结果中剔除的方法来达到“伪删除”的效果，但随着删除数据的增多会使得用户的使用体验变差。因此图片删除功能的实现需要从 iSimilar 核心上进行改进，可通过为 iSimilar 图片仓库增加删除操作，或在检索过程中综合已删除图片的 ID 进行筛选。

（5） 数据合并方法

由于各网站的网页结构、层次、组织结构不同，对提取出来的数据的分析和合并处理也相应有所不同，这给系统的数据准备工作带来了一定的困难。对于使用基于 XPATH 的模板提取方法得到的数据，可考虑使用模板进行逆向分析来完成数据合并。

iSearch 广告信息增强系统的下一步工作将围绕上述问题展开，重点集中在提高数据的实时性、完整性、增量索引的改进及图片删除这几个方面。

# 参考文献


[1] 刘永．总局召开广告业发展座谈会[N]．中国工商报．2013-03-19（1）

[2] J. A. García*，Rosa Rodrigue-Sánchez，J. Fdez-Valdivia，J. Martinez-Baena．Comparative visibility analysis of advertsement images[J]．Signal Processing: Image Communication．2011．26：580-611

[3] 中国互联网络信息中心（CNNIC）．第 31 次《中国互联网络发展状况统计报告》[OL]．文献网址：http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201301/P020130122600399530412.pdf．2013-01-15

[4] 顾其威，郭鹏，潘锋．手机广告推荐中的用户兴趣建模研究[J]．计算机应用研究．2012-02．29（2）：579-585

[5] Jung woo Lee，Choong sik Lee，Yong suk Park．Research on the Advertisement Effect of Push Type Mobile Advertisement[C]．Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology．北京．中国．1 月．2009 年．27

[6] Y Rui，S H Thoma，S F Chang．Image retrieval: Past, present, and future[J]．Journal of Visual Communication and Image Representation．1999．10（1）：39-62

[7] Th. Hermes，Ch. Klauck，J. Krey，J. Zhang．Content-based Image Retrieval [C]． Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research．1995．30~143

[8] Niblack.W.，Barber.R.，Equitz.W，Glasman E.，Petkovic D.，Yanker P.，Faloutsos C.，Taubin G.．The QBIC project: Querying images y content using color, texture, and shape[C]． Proc SPIE, Storage and Retrieval for Imaging and Video Databases．1993．1908．173-187

[9] 李向阳，庄越挺，潘云鹤．基于内容的图像检索技术与系统[J]．计算机研究与发展．2001．38（3）：344-354

[10] Jeffrey R. Jeffrey，Charles Fuller，Amarnath Gupta，Arun Hampapur，Bradley Horowitz，Rich Humphrey，Ramesh C. Jain，Chiao-Fe Shu． Virage image search engine: An open framework for image management．Proc SPIE: Storage and Retrieval for Still Image and Vedio Database IV．1996．2670：76~87

[11] Smith.J.R，Chang.S.F．Local color and texture extraction and spatial query．International conference on image processing．1996．3：1011~1014

[12] Mahmood R.，Azimi-Sadjadi，Jaime Salazar，Saravanakumar Srinivasan．An Adaptable Image Retrieval System With Relevance Feedback Using Kernel Machines and Selective Sampling．IEEE Transactions on Image Processing．2009．18（7）：1645-1659

[13] Azadeh Kushki，Panagiotis Androutsos，Konstantinos N. Plataniotis，Anastasios N. Venetsanopoulos．Query Feedback for Interactive Image Retrieval．IEEE Transactions on Circuits and Systems for Video Technology．2004．14（5）：644-655

[14] Paisarn Muneesawang，Ling Guan．An Interactive Approach for CBIR Using a Network of Radial Basis Functions．IEEE Transactions on Multimedia．2004．6（5）：703-716

[15] David Picard，Arnaud Revel，Matthieu Cord．An application of swarm intelligence to distributed image retrieval[C]．Information Sciences．2012．192：71-81

[16] Gwénolé Quellec，Mathiue Lamard，Guy Cazuguel，Béatrice Cochener，Christian Roux．Adaptive

Non-Seperative Wavelet Transform via Lifting and its Application to Content-Based Image Retrieval[J]. IEEE Transactions on Image Processing. 2010. 19（1）: 25-35
[17] David G. Lowe. Object Recognition from Local Scale-Invariant Features[C]. The Proceedings of the Seventh IEEE International Conference on Computer Vision. 1999. 2: 1150-1157
[18] A. Oliva, A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope[J]. International Journal of Computer Vision. 2001. 42(3): 145–175
[19] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity Search in High Dimensions via Hashing[C]. Proceedings of the 25th International Conference on Very Large Data Bases. San Francisco. USA. 1999: 518-529
[20] MI Mauldin. Lycos: Design choices in and Internet Search Service[C]. IEEE Expert. 1997. 12（1）: 8-11
[21] IA（互联网档案馆）组织的开源爬虫项目 Heritrix 相关介绍，http://crawler.archive.org/
[22] Nutch 项目介绍（Apache 公司官方网站），http://nutch.apache.org/
[23] Tom White. Hadoop: The Definitive Guide. 3rd Revised edition. USA. O'Reilly Media. 2012
[24] Dhruba Borthakur. HDFS Architecture Guide[OL]. 文献网址：http://archive.cloudera.com/cdh4/cdh/4/mr1/hdfs_design.pdf. 2008
[25] Tika 介绍（Apache 公司官方网站），http://tika.apache.org/
[26] Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters[J]. Communications of the ACM. 2008. 51: 107-113
[27] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, Khaled Shaalan, A Survey of Web Information Extraction Systems[J]. IEEE Transactions on Knowledge and Data Engineering. 2006. 18: 1411-1428
[28] Yanhong Zhai, Bing Liu. Web Data Extraction Based on Partial Tree Alignment[C]. Proceedings of the 14th international conference on World Wide Web. New York. USA. 2005: 76-85
[29] Xpath 介绍，http://www.w3school.com.cn/xpath/
[30] Otis Gospodnetic, Erik Hatcher 著，谭鸿，黎俊鸿，周鹏，高承山译. Lucene in Action（中文版）. 电子工业出版社. 2007
[31] Mathias Lux, Savvas A. Chatzichristofis. LIRe: Lucene Image Retrieval – An Extensible Java CBIR Library[C]. Proceeding of the 16th ACM international conference on Multimedia. New York. USA. 2008: 1085-1088
[32] 李刚. 轻量级 Java EE 企业应用实战：Struts2+Spring3+Hibernate 整合开发. 第 3 版. 电子工业出版社. 2012
[33] Scott Raymond. Ajax on Rails. 影印版. 东南大学出版社. 2007

# 致　谢

感谢我的论文导师朝红阳教授，在我完成论文的过程中，给予了悉心的指导。朝教授一直教导我们要有严谨求实的态度，叙述要有逻辑性，结论要有实际的理论和实验支撑；这不仅对我的学业有很大帮助，也对我的人生有着很大的启发。

同时，要感谢 iSimilar 图像检索平台提供的图像检索技术支持。此外，我要感谢丁圣勇和丁剑冰师兄，他们不仅帮助我分析在系统设计和实现的过程中遇到的问题，还给予了强大的技术及实验环境支持，使得我的论文能够顺利完成。

最后要感谢我的家人和朋友，在学习、生活、精神上给予了我支持和鼓励，陪我渡过了一个个难关。