# 面向社区的疾病诊断决策支持系统的设计与开发

摘要：

近年来，我国大力发展社区医疗服务，社区医疗已经成为解决我国人民看病难、看病贵等问题的一个重要方式。社区医疗服务是医疗体制改革和社区建设的重要组成部分，是提供居民的基本社区医疗服务。其优点是方便就医、减少病人的诊疗费用和时间，是提高居民的健康水平的重要保障。

然而，但由于受经济发展水平、医护人员素质、政策扶持等诸多因素影响，使得社区医疗服务的整体水平偏低。随着居民健康意识的不断提高，对医疗质量的要求也在提高。社区医疗无法满足居民的医疗需求，造成了医疗服务集中于大医院，而社区医疗闲置的现状。目前我国各地社区医院普遍存在基础医疗设施差和医护人员诊疗水平低的客观事实，因此如何改善这一现状是国家和地方社会保障和医疗服务体系重点关注的问题之一。

大量研究表明，临床决策支持系统的应用可以有效解决临床医生知识的局限性问题、减少人为疏忽、相对降低医疗费用，为医疗质量提供了保障。因此在社区医疗服务中，充分发挥信息技术的优势，利用临床决策支持技术，可以提高社区医疗服务水平，解决目前医疗资源配置不均问题。

本文首先通过分析社区医疗需求得到面向中心医院、社区医院、患者的三级决策支持服务模式，并基于此服务模式，以疾病诊断为切入点，分析面向社区的临床决策支持系统。由于社区分布的广域性以及疾病种类的多样性，。因此设计具有开放性和可扩展性的系统框架，支持高速增长又具有多样性的医疗数据的管理以及多种疾病的诊断的决策支持服务的添加和更新,是非常有必要的。本文围绕该目标，主要从以下几点展开：

* 首先通过分析社区医疗的存在的诊疗水平低的问题，提出将临床决策支持系统应用于社区医疗，进一步分析社区决策支持的服务模式，并基于此模式，以疾病诊断为切入点，研究构建满足社区医疗需求的决策支持系统的框架
* 分析并解决框架的关键性技术问题，开发实现系统框架
* 以头痛和老年痴呆症为例开展基于框架的系统实践，开发原发性头痛诊断决策支持系统和老年痴呆症诊断决策支持系统，验证系统框架具有一定的开放性和扩展性，能够支持不同疾病的诊断决策支持系统的快速开发和统一管理。

## 第1章 引言

### 1.1课题背景

社区医疗服务为居民提供基本的医疗服务，是我国医疗体制改革和社区建设的重要组成部分。它是以人的保健为中心、家庭为单位、社区为范围导向，以妇女儿童、老年人、慢性病、残疾人和脆弱人群为重点，以解决社区主要问题、满足社区基本卫生需求为目的，融预防、医疗、保健、健康教育为一体的，有效、经济、综合、连续的基层医疗服[[1](#_ENREF_1)]。

自1997年国务院提出要大力发展社区医疗服务以来，全国社区医疗服务机构的数量逐渐增多，网络布局也逐步合理，至2011 年底，我国已建立社区医疗服务中心（站）32860个，其中，社区医疗服务中心7861个，社区医疗服务站24999个[[2](#_ENREF_2)]。社区医疗服务在一定程度上缓解了“看病难，看病贵”的难题，给人们就近就医带来了不少方便，但是其成效并没有达到事先预想的居民“大病去医院、小病到社区”。据不完全统计，目前我国还有1/4 的病人患病没有就医，虽有部分病人到药店购药治病，但超过5 成以上的病人还是到三级大医院就医[[3](#_ENREF_3)]。2010年全国医疗服务情况显示，全国社区卫生服务中心病床使用率为56.1%，乡镇卫生院为59%，而三级医院和二级医院分别为102.9%和87.3%[[4](#_ENREF_4)]。根据卫生部数据显示，2012年基层医疗卫生机构门诊量增加3.0亿人次，增长7.9%。由于医院门诊量增速高于基层医疗卫生机构，基层医疗卫生机构占门诊总量的比重下降1.1个百分点。

由于我国的医疗服务过多地集中于上级医院而不能向基层合理分流，为了引导患者在社区就诊，合理利用医疗资源，2008年我国卫生部规定了社区卫生服务机构实行首诊负责制和双向转诊制。实施社区首诊制，有利于促进患者的合理分流，使一些常见病可以在社区得到解决，可以缓解看病难的问题。同时可以使大医院有更多的精力专注于疑难杂症、危重病的诊疗，提高卫生资源的利用率，促进基层医院与大医院共同发展，但首诊制的具体实施仍存在困难，主要原因在于社区居民对社区卫生服务机构的医疗水平和设备缺乏信心[[5](#_ENREF_5)]。卫生部的资料表明，三级医院65% 的门诊病人和77%的院病人均可分流到社区服务中心，但仅有22.5%的人愿意去社区医院就诊，分析原因主要是患者认为社区医疗水平有限[[6](#_ENREF_6)]。可见目前我国社区医院医护人员诊疗水平低是制约医疗资源合理利用的关键因素。

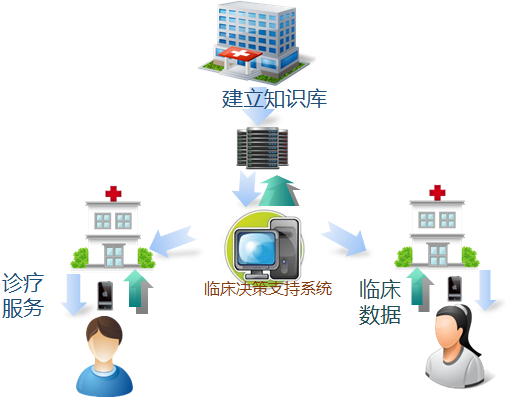
国外对社区卫生建设的理论与实践研究起步较早。1957年，联合国在发达国家开始倡导社区卫生发展工作，澳大利亚、新西兰、英国、美国、加拿大等是社区卫生发展最早的国家。英国是现代社区卫生服务的发源地，经过几十年的发展，形成了全民免费的国家保健服务系统，医疗卫生信息服务非常发达，拥有各级各种专业化网站，如社区卫生服务合作交流网站等，便于医生、护士了解患者信息及卫生资源，从而做出更及时、更准确的诊断[[7](#_ENREF_7)]。美国社区卫生服务起源于19世纪末，社区医院及其他社区卫生服务机构形成连锁的组织，资源的使用效率得到了提高。这些国家在各自社区建设发展中分别形成了适合自身经济、社会和历史文化特征的社区治理方式，使城乡社区服务得到有序协调发展。我国社区卫生服务应借鉴和吸收国外先进经验和做法，大力完善社区卫生服务体系，提高社区卫生服务质量。

临床决策支持系统作为帮助医生做出更好的临床决策的工具，已经有大量研究表明它具有提高医疗质量和安全性的作用，有很多对于前期研究的总结如 2006，1944年、1998、2005 、2005根据最近一份2005 年发表于JAMA 的比较权威的系统性回顾[4]，说明大多数国外报道的临床决策支持系统对医生的工作质量有提高作用。这些综述都展现了临床决策支持系统在诊断、用药和预防方面很好的效果。（扩）

如果能利用临床决策支持服务于社区医疗领域，将常见多发疾病的临床知识整合到一起，有针对性地及时提供给临床医生，辅助他们形成最终诊疗决策，就可以减少医疗差错，提高社区医疗质量和效率。根据2007发表在JAMIA的白皮书—A Roadmap for National Action on Clinical Decision Support[[8](#_ENREF_8)]，虽然临床决策支持系统在某些医疗机构对于病人的诊疗工作已经产生了很好的效果，但是在很多其他的医疗机构，CDS并没有投入实际应用。为了CDS更广泛地应用于临床医疗，该文指出，必须通过三大支柱实CDS的目标：1.Best knowledge Available When Needed;2.High Adoption and Effective Use;3.Continuous Improvement of Knowledge and CDS。

根据这个原则，决策支持系统需要将临床专家或者已有的知识提供给社区医生，而社区医生通过获取决策支持的服务，给患者提供更好的医疗服务，在这个过程中患者的数据最终汇集为临床数据，为临床专家提供临床研究分析的数据来源，从而对于已有的知识进行更新，更新之后系统也随之进行更新，从。

由分析得到在社区应用决策支持系统的服务模式如下：



通过此服务模式，由中心医院负责知识的表达与更新，通过决策支持系统为社区提供诊疗辅助的服务，社区医生在需要的时候获取决策支持服务，而知识本身也在不断的更新中。从临床角度来看，社区医生在需要获取知识的时候通过系统获取了当前由专家提供的最新的知识，提高了医疗水平，保障了医疗质量；从科研角度来看，系统提供便捷的病历存储、检索和浏览功能，领域专家可以及时准确统计、分析和处理临床信息，从而发现问题和获取新的知识，通过更新决策支持系统的知识库，进一步修正和完善疾病的诊疗规范。

在此模式中，最为关键的部分为临床决策支持系统。而社区医疗主要目标是满足常见病的诊疗需求，因此提高社区医生常见病的诊断水平是提高社区医疗水平的重要环节，所以构建面向社区的临床决策支持系统是解决目前社区医疗问题的可靠途径。

### 1.2面向社区的疾病诊断决策支持系统

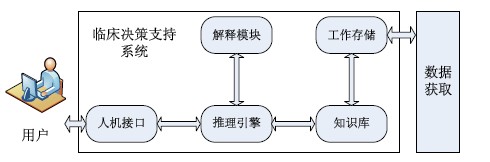
本节主要对于面向社区的疾病诊断决策支持系统的设计思路进行介绍。通过分析社区临床对于临床决策支持系统的需求，参照已有的临床决策支持系统架构，提炼系统面临的关键问题，形成面向社区的疾病诊断决策支持系统的框架。

#### 1.2.1 临床诊断决策支持系统概述

关于临床决策支持系统的定义有很多种说法。最为广泛接受的是：临床决策支持是通过组织性的相关的医学知识和病人信息来提高医疗决策的行为从而改善医疗服务的一种过程[[9](#_ENREF_9)]。CDS典型的应用包括药物互斥作用的警报、电子化的剂量提示和基于临床指南的医嘱集。而诊断决策支持系统可以引导医生得到正确的诊断，并且能够减少用药的错误。第一代的诊断决策支持的产品(例如, QMR—First Databank, Inc, CA;Iliad—University of Utah; DXplain—Massachusetts GeneralHospital, Boston, MA)使用基于疾病的特征性症状、体征和生理实验检查结果的预编译知识。用户会让自己的病人从选项菜单中选择输入的结果，并且这些程序将使用贝叶斯逻辑或模式匹配算法，提出诊断可能性。通常情况下，系统提出的建议是具有临床价值的，而且也对于临床推理教学方面有一定的帮助[[10](#_ENREF_10), [11](#_ENREF_11)]。

从20世纪70年代，到现在，决策支持系统研究经过了五十多年的发展，积累了大量的研究成果，特别是在各种疾病的诊疗方面：Jerick等人开发了诊断肺病的临床决策支持系统，Delphi与美国糖尿病协会（ADA）合作开发的Delphi糖尿病管理软件。哈佛医学院开发的“DXPI．AIN”系统包含2200种疾病和5000多种症状，针对某一种疾病的专项医学专家系统更是举不胜数。Umbau开发了皮肤癌辅助诊断系统。Prov等人研制了用于诊断慢性腹痛的决策支持系统。2000年wells等人开发了计算机辅助乳腺治疗计划系统。上个世纪80年代以来国内涌现了一批专科临床决策支持系统，如肝病营养疗法专家系统、昏迷诊断计算机专家系统、急性肾衰诊断系统、颈腰疾病专家系统、精神疾病诊断系统、心功能辅助诊断系统、针灸专家系统及医病诊疗用药系统。这些系统经过临床验证都对于疾病的诊疗有积极的效果。

临床决策支持系统大多数由专家系统而来，因此典型的系统结构如下：



临床决策支持系统一般由推理引擎、知识库、解释器、工作存储以及人机交互五个部分组成。

知识库（Knowledge base, KB）是医学决策支持系统的基础，用于存放各种专家诊断知识，包括医学事实和可行的操作与诊断规则等。医学推理引擎所用的知识来源于知识库。知识是决定一个决策支持系统性能是否优越的主要因素，而知识的表达又是决定知识可用性的重要一环。

推理引擎（Inference Engine，IE）是知识规则应用于问题求解的载体，根据 全局数据库的当前内容，利用知识库中的规则，按照一定的推理策略，去解决当前的问题。

工作存储（Working Memory, WM）用于存放决策所需的病人的初始数据，如出生年月、性别等人口学指标、过敏史、使用中的药物，中间推理数据和诊断结果及一些对结果的诊断处理等。

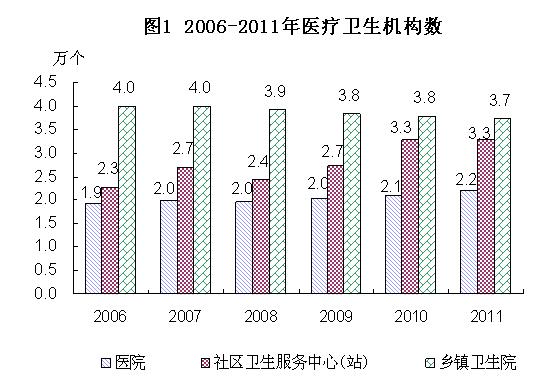
人机接口（Man-Machine interface）是系统与用户进行对话的界面。用户通 过人机接口输入必要的数据和信息、提出问题和获得系统对问题的推理结果以及 获得系统对结果或用户提问所做出的解释；系统则通过人机接口要求用户回答系统的询问。

解释模块（Explanation Module）负责向用户解释推理的过程，如诊断结论是如何得出的。解释模块具有非常重要的作用，但是不是所有的决策支持系统中都存在此模块。

#### 1.2.2 关键技术问题

由于社区医疗的特殊性，面向社区疾病诊断决策支持系统需要解决以下问题：

* 系统的用户数量规模大。随着生活水平的不断提高，人们对于医疗的需求量也在不断增高。自2008年以来我国社区医疗机构数目在不断增长中，而社区医疗服务人员也在不断增加，至2011 年底，我国已建立社区医疗服务中心（站）32 860个，社区医疗服务中心人员32.9 万人，社区医疗服务站人员10.4 万人。

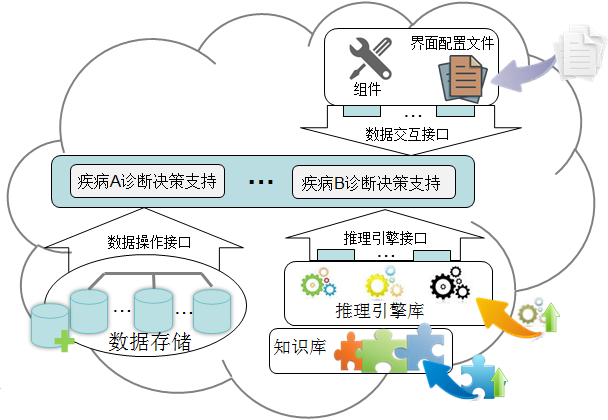


* + - 系统的数据存储量庞大。由于社区覆盖的范围广，人口基数庞大，每天在社区诊疗中将产生大量的医疗数据，这些数据包含了不同科室的治疗、检查检验和护理等信息，如医嘱单、处方、检查检验报告、病历和病程记录以及其他诊疗相关的信息，涵盖了病人整个诊疗过程中产生的信息。这些数据种类多样、格式复杂，而且存储量巨大。
    - 系统的决策支持的内容涵盖常见疾病。由于社区医疗的职责在于对于居民的一般的常见病、多发病进行首诊的工作，面向社区的疾病诊断决策支持系统需要提供覆盖常见疾病的决策支持服务。但是不同疾病的诊疗过程有着巨大的差异，例如头痛诊断主要依据患者对于头痛症状的主观描述，如头痛时间、头痛程度，而老年痴呆症的诊断是通过标准化量表来评估患者的记忆力等因素再综合考虑。

如果从典型的临床决策支持系统的结构出发，构建面向社区的疾病诊断决策支持系统，很难满足以上需求。因此，对于面向社区的疾病诊断决策支持系统，必须建立一种可扩展、具有开放性的系统架构，支持高速增长又具有多样性的医疗数据的管理以及多种疾病的诊断的决策支持服务的动态添加和更新。

#### 1.2.3 系统技术框架设计

基于以上分析，论文形成了面向社区疾病诊断决策支持系统的框架的研究目标，框架的概念图如下：



通过对于系统框架中关键技术问题进行研究，实现系统框架，通过选取典型病种对于系统框架进行验证。

### 1.3研究目标与内容

随着我国经济水平的发展，人民生活水平的提高，人们对于健康的需求也越来越高。近年来，我国大力发展社区医疗服务，希望通过社区医疗服务为居民提供基本的医疗服务，开展社区首诊制，引导居民“ 小病进社区，大病进医院”，可以促进患者合理分流，合理利用医疗资源，控制医疗费用过快增长，缓解“ 看病贵、看病难”问题。同时可以使居民享受到稳定、持续的医疗服务，从而更好地保障居民的健康。但是由于我国的社区卫生服务起步相对较晚，发展相对滞后，患者仍集中于上级医院就医，造成上级医院医疗负担过重、超负荷运作，而社区医疗资源闲置浪费。经研究调查分析得到其中多数患者不愿到社区医疗机构就医的原因在于社区医疗水平较低，无法满足日益提高的医疗需求。大量研究表明临床决策支持系统应用于临床能够提升医疗质量，降低医疗差错。目前国内外已经研制了许多针对专科疾病诊断进行决策支持的系统，在临床试验中也显示了良好效果，然而广泛应用于临床日常使用的并不多。结合国外对于应用CDS的建议以及社区医疗的需求，分析得到中心医院-社区医生-患者的三级决策服务模式。并以社区的疾病诊断决策支持系统为出发点，研究系统面临的关键问题，并以此构建可扩展、开放性的社区疾病诊断决策支持系统框架，为社区疾病诊断系统提供了构建的方案。

研究内容包括：分析社区医疗问题，提出满足社区医疗需求的临床决策支持系统的服务模式，基于此模式分析研究社区疾病诊断决策支持系统的关键问题，针对以上关键问题，设计面向社区的临床决策系统的总体架构。以头痛和老年痴呆症两种疾病为例，设计并实现原发性头痛的临床诊断决策支持系统和老年痴呆症诊断决策支持系统，并对系统框架的可扩展进行验证和讨论。

### 1.4 章节编排

本文一共分五章对面向社区的疾病诊断决策支持系统进行了研究，文章结构如下：

第一章：简述了研究背景，介绍研究目的与意义，最后提出了构建具有开放性可扩展的面向社区的疾病诊断决策支持系统的框架

第二章：研究面向社区疾病诊断决策支持系统的框架设计的关键问题与技术，简介框架的实现和基于本框架的疾病诊断决策支持系统的开发方案

第三章：选取头痛作为目标疾病，基于框架开发原发性头痛诊断决策支持系统，并分析验证系统框架

第四章：选取老年痴呆症作为目标疾病，基于框架开发老年痴呆症诊断决策支持系统，并分析验证系统框架

第五章： 总结全文，提出展望

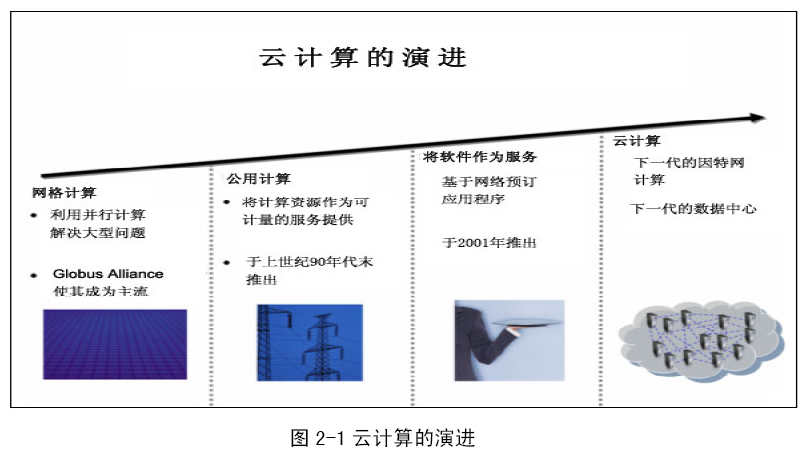
## 第2章 系统框架设计与实现

### 2.1 相关技术概述

#### 2.1.1 云计算

##### 2.1.1.1 云计算定义

云计算是在分布式计算（Distributed Computing）、网格计算（Grid Computing）和并行计算（Parallel Computing）基础上发展起来的，其前身是利用并行计算解决大型问题的网格计算和将计算资源作为可计量服务提供的公用计算，在互联网宽带技术和虚拟化技术高速发展后发展起来【】



由上图我们可以看出，云计算其实并不是新技术革命性的发展，它是分布式计算、网格计算、并行计算、效用计算、平台即服务、软件即服务、基础设施即服务等概念综合演进的结果。云计算的概念现在还没有统一的定论，不同的人对云计算的定义和理解也不相同。

维基百科给云计算下的定义为：云计算是将IT 能力以服务的方式提供给用户，允许用户在不了解提供服务的技术、没有相关知识以及设备操作能力的情况下，通过Intemet 获取需要的服务

云计算专家刘鹏认为：云计算是一种商业模型，它将计算任务分布在大量计算机构

成的资源池上，使用户能够按需获取计算力、存储空间和信息服务[4]。

IBM 认为：云计算是一种计算模式，在这种模式中，应用、数据和IT 资源以服务的方式通过网络提供给用户使用。云计算是一种基础架构管理的方法论，大量的计算资源组成IT 资源池，用于动态创建高度虚拟化的资源方式提供给用户使用。

因此，这些资源被比喻为“云”。“云”就像是一个发电厂，只是它提供的不是电力，而是计算机的计算、应用和管理能力。只要通过网络连接，并得到授权，就可以使用这些能力和资源[5]。

1)能够整合大规模异构计算资源

传统的分布式计算一般仅能应用于一个小范围的计算网络(如局域网)，且对计算资源的同构性要求较高，难以处理在计算和存储能力、操作系统、开发平台等方面存在很大差异的计算资源。而通过云计算则可以整合分布在一个广阔地域内的、分属于若干个组织的计算资源，形成一个功能非常强大的计算和存储平台。另外，云计算并不要求计算设备在硬件或软件上具有很强的共性，绝大部分计算设备都可以被整合成为云计算平台的一部分。的升级一般仅需要几分钟H]，且可以在不影响系统整体运行的情况下动态进行。此外，云计算平台可以建立在现有的硬件基础上，在升级时也只需按照需求增添相应的设备，而不需要像升级传统计算平台那样将设备完全更换，从而可节省大量硬件购置成本。

2)易于动态扩展

可扩展性是云计算与传统计算模式相比的最大优势之一。由于云计算能够集成硬件种类、网络类型、操作系统、软件平台等各不相同的各种计算设备，因此，在需要时云计算平台的计算和存储能力可以得到方便和快速的扩展。与传统计算平台需要几天甚至几个星期的系统升级时间相比，云计算平台的升级一般仅需要几分钟H]，且可以在不影响系统整体运行的情况下动态进行。此外，云计算平台可以建立在现有的硬件基础上，在升级时也只需按照需求增添相应的设备，而不需要像升级传统计算平台那样将设备完全更换，从而可节省大量硬件购置成本。

3)虚拟化与服务

虚拟化也是云计算的一个重要特征。无论一个云计算平台实际整合了多少计算设备，在用户看来其就是一个单一实体，也是获得计算服务的唯一接口。由于应用了虚拟化技术，云计算平台既可以将多个计算任务放在同一台功能强大的设备(如大型工作站)上运行，也可以将一个计算任务拆分成若干部分，分别在多台设备上运行。这样，就可以最大限度地利用系统内的闲置计算资源。此外，通过利用虚拟化技术，云计算平台可以根据客户的需求动态分配计算资源和构造系统平台。此外，若干设备的故障不影响云计算平台整体运行，也不会中断向用户提供服务。

4)以Internet为基础的通信平台

与传统分布式计算不同，云计算通过Internet进行各个设备之间的通信。由于Internet已经有了非常成熟的标准、体系和技术，这在很大程度上保证了云计算系统通信的可靠性和安全性。此外，由于云计算建立在国际通行的通信协议的基础上，这使得其易于与各种流行的软件开发技术集成

##### 2.1.1.2云计算平台应用

从体系结构的角度来看，一个云计算系统是为配置各种计算机软件、硬件和网络资源、运行客户应用程序而搭建的平台

根据云计算服务的所在层次，可分为以下几种云计算服务类型：

1．基础设施即服务(Infrastructure as a Service，IaaS)。IaaS分为资源集和基础设施服务。资源集在最底层，具体可分为物理资源和虚拟资源。通过虚拟化技术，使云计算能够对物理硬件像虚拟资源一样的自动进行管理，也能使用通用的资源接口对计算资源和存储资源进行虚拟化管理。基础设施服务具体分为基本基础设施服务和高级基础设施服务，基本基础设施服务又分为计算服务、存储服务和网络服务。

2．平台即服务(Platform as a Service，PaaS)。PaaS是将软件开发环境以服务方式在云计算环境里提供，具体分为设计环境和执行环境。设计环境有Caroline和Django，执行环境有Google App Engine和Azure。

3． 软件即服务(Software as a Service，SaaS)。SaaS分为基础软件服务和复杂软件，复杂软件服务一般由基础软件服务通过服务组合、集成等方式形成。运行于云平台上的全部应用软件都属于SaaS，它直接为用户提供各种各样的服务。

主流的PaaS提供商有：

#### 2.1.2 NoSQL数据库

##### 2.1.2.1 NoSQL数据库概述

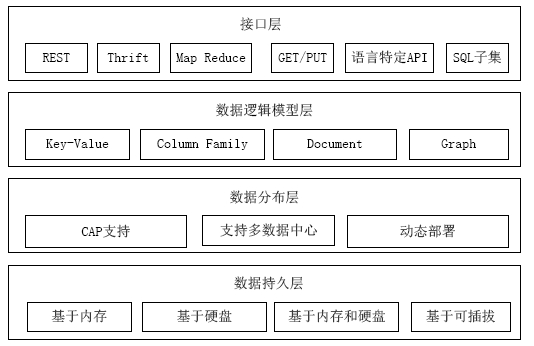
字面上的NoSQL是两个词的组合：No和SQL,它暗示着NoSQL技术/产品与SQL之间的对立性。其实这个词在早期刚出来的时候可能的意思是No RDBMS（非关系型数据库管理系统）或者No relation（非关系型）。后来随着NoSQL的发展，人们又提出了NoSQL是NotOnly SQL（不仅仅是SQL）的简称。不管字面上的意思如何，今天NoSQL泛指这样一类数据库和数据存储，他们不遵循经典的RMDBS原理，且常与Web规模的大型数据库相关。换句话来说，NoSQL不是单指一个产品或者一种技术，而它代表的是一族产品，以及一系列不同的、有时候相互关联的、有关数据存储及处理的概念。NoSQL数据库不需要特定的表结构，通常不支持表的连接操作，不支持完整的ACID属性，而且一般拥有强大的可扩展性。下面是Sourav Mazumder对NoSQL数据库的一个比较严谨的描述[19]：

（a）不像关系数据库会建立一个固定的关系模式，NoSQL会建立一个松散的、易扩展的数据模型来进行数据建模，比如文档、列形式。

（b）系统设计会按照CAP[20]原则，在跨多个节点之间建立数据分布式模型，使系统具有较强的水平扩展特性。

（c）有在磁盘以及内存中持久化数据的能力。

（d）进行数据访问的时候支持多种NoSQL接口。



在前面讲过，NoSQL并不是单指一个产品或者一种技术，而它代表的是一族产品。在这个NoSQL产品族里面，拥有数量巨大的满足各种需求的非关系数据库。通常情况下，对NoSQL数据库进行分类，主要是看它们数据存储的方式，例如键值存储、图形数据库、文档存储等等[21]。

（1）面向列的有序存储

面向列的数据存储例如Google Bigtable，HBase的数据存储模型支持面向列，这与RMDBS面向行的存储格式截然不同。面向列的存储能高效的存储数据，如果列值不存在，就不存储，这样一来，遇到null值的时候，就可以避免存储空间的浪费。 每个数据单元都可以看做一组键/值对集合，单元本身通过主标识符来辨别，这个主标识符就是主键，Bigtable与其他类似的数据叫做行键（row-key）。在实际的数据存储中，物理上的一条数据的列族并不互相隔离，同一行健的所有数据存储在一起，列族可以看做代表成员列的键，行健代表整条数据的键。在BigTable和其他类似的列族数据库中，数据按顺序连续存储。当数据逐渐填满一个节点之后，它会拆分成多个节点。数据不仅在每个节点上是有序的，而且还跨越多个节点成为一个更大的有序集。在数据持久化存储方面会有容错的考虑，每份数据都同时维护三个副本。大部分类BigTable产品都利用分布式文件系统将数据持久化存储在磁盘上。分布式文件系统支持将数据存储到集群的多台服务器上。 因为有序，数据按行健查找效率极高。数据访问随机性小，查找也简单，就是在序列中查找包含数据的节点。数据插入发生在数据队列的尾部，数据更新则原地进行，不过一般不是原地覆盖写入，而是添加一个数据的新版本到指定的单元里面去。每个单元始终维护多个版本，版本属性通常可以配置。 列族数据库的代表为：Bigtable以及HBase

（2）键/值存储

哈希表（HashMap）或者关联数组是可以容纳键值对的最简数据结构。这类数据结构非常受欢迎，因为他们极其高效，访问数据的时间复杂度为O（1）。键值对中的键在集合中是唯一的值，容易查找，便于访问数据。 键值存储各不相同：有的数据保存在内存中，有的能把数据持久化写入到磁盘里。键值对可以被分散保存到集群节点中。 像Oracle的Berkeley DB键值存储非常强大，它是纯粹的存储引擎，键和值都是字节数组，其核心存储引擎并不关注键或值的意义，只管保存传入的字节数组对，然后返回同样的数据给调用的客户端。Berkeley DB支持将数据缓存在内存中，随着数据的增长，将其拷贝在磁盘里。

另一种键值存储是缓存。缓存提供应用中使用最多的数据的内存快照。缓存的目的是减少磁盘的I/O。它可以是最简单的映射表，也可以是支持缓存过期策略的健壮系统。作为一种流行策略，缓存广泛应用于计算机软件栈所有层面以提高性能。操作系统、数据库、中间件和各种应用都使用缓存像EHCache这样的健壮的开源分布式缓存系统广泛的应用在各类Java应用中，可以将它看做一种NoSQL方案。另一种缓存系统Memcached在Web应用中非常流行，它是开源的高性能对象缓存系统。

（3）文档数据库

文档数据库不是文档管理系统。文档数据库中的文档一词指的是文档中松散结构的键/值对集合通常是JSON，而非一般意义的文档或表格。 文档数据库将文档当做一个整体，不会将文档分割成多个键/值对。在集合的层面上，这意味着可以将不同结构的文档放在同一个集合里。文档数据库支持文档索引，不仅包括主标识符，还包括文档的属性。现在为数不多的文档数据库中，其中最有名的还是MongoDB和CouchDB。

（4）图形数据库

这种数据库设计用来保存关系图（图中连接的元素有数目不详的关系）。这种数据图可能是一种社交关系图、公共交通图、道路图或者拓扑图。

##### 2.1.2.2 MongoDB简介(传感器数据)

MongoDB[24][25]是文档存储，文档按组分成集合。在概念上，可以认为集合类似关系表。不过集合不对schema进行严格约束，这点和关系数据库的表不同。一个集合可以包含任何文档，不过为了便于有效快速地检索，同一个集合中文档还是相似的比较好。

MongoDB数据库主要特性: （1）文档数据类型 文档存储为BSON格式。BSON是JSON的二进制编码形式，结构类似嵌套键/值对。BSON是JSON的超集，额外支出一些类型，比如正则表达式、二进制数据和日期。每个文档都有唯一的标示符，如果数据在插入文档之前没有添加唯一标示符，则由MongoDB自动生成一个标示符ID。类似的存储格式如下： {\_id:ObjectID:(„0d09a7c7d7a812a54c74d984a‟), Title: „„„‟, Url:‟„„‟ } 表示的是一个文档中的存储数据。 （2）即时查询 说一个系统支持即时查询即是无需预先定义系统接收的查询类型。关系数据库用这个能力，他们会严格遵照指示执行任何完备的sql查询，无论有多少条件。虽然这个在关系数据库里面是理所当然的功能，但是在NoSQL中并非所有都支持动态查询。比如键值存储数据库只能按照一个维度-键来进行查询。 （3）二级索引 MongoDB中的二级索引是用Bshuttle（B-Tree）实现的，B树索引也是绝大数关系数据库支持的索引，针对多种查询进行优化 （4）复制 MongoDB通过称为副本集（replica set）的拓扑结构提供复制功能。副本集将数据分布在多台机器上以实现冗余，在服务器和网络故障的时候，能提供自动故障转移。除此之外，复制功能还能用于扩展数据库的读能力。如果有一个读密集型的应用程序，可以把数据库的读操作分散到副本集群中的各台机器上。 （5）速度和持久性 在MongoDB中，用户可以选择写入语义，用户决定是否开启Journaling日志记录，通过这种方式来控制速度和持久性间的平衡。该日志记录将所有的写操作都提交到一个日志中，即使是服务器非正常关闭（比如断电故障），该日志也能保证重启服务后，恢复之前的写操作。 （6）数据库扩展

MongoDB的水平扩展非常易于管理，它通过基于范围的分区机制，即自动分片来实现这一设计目标，自动分片机制会自动管理各个节点之间的数据分布。分片系统会自动处理节点的增加，帮助进行自动故障转移

#### 2.1.3 Web Service

Web Service也叫XML Web Service Web Service是一种可以接收从Internet或者Intranet上的其它系统中传递过来的请求，轻量级的独立的通讯技术。通过SOAP在Web上提供的软件服务，使用WSDL文件进行说明，并通过UDDI进行注册。它包含以下关键的技术：

XML：(Extensible Markup Language)扩展型可标记语言。面向短期的临时数据处理、面向万维网络，是Soap的基础。

Soap：(Simple Object Access Protocol)简单对象存取协议。是XML Web Service 的通信协议。当用户通过UDDI找到你的WSDL描述文档后，他通过可以SOAP调用你建立的Web服务中的一个或多个操作。SOAP是XML文档形式的调用方法的规范，它可以支持不同的底层接口，像HTTP(S)或者SMTP。

WSDL：(Web Services Description Language) WSDL 文件是一个 XML 文档，用于说明一组 SOAP 消息以及如何交换这些消息。大多数情况下由软件自动生成和使用。

UDDI (Universal Description, Discovery, and Integration) 是一个主要针对Web服务供应商和使用者的新项目。在用户能够调用Web服务之前，必须确定这个服务内包含哪些商务方法，找到被调用的接口定义，还要在服务端来编制软件，UDDI是一种根据描述文档来引导系统查找相应服务的机制。UDDI利用SOAP消息机制（标准的XML/HTTP）来发布，编辑，浏览以及查找注册信息。它采用XML格式来封装各种不同类型的数据，并且发送到注册中心或者由注册中心来返回需要的数据。

Web Service中的角色及其相互关系

在Web Service架构中，首先定义了三耱角色：服务提供者(service

provider)，服务注册处(service registry)或者叫注册中心和服务请求者

(service requestor)，分别说明如下：

(1)服务提供者

从企业的角度来看它是服务的所有者，从整个体系的角度看它是容纳

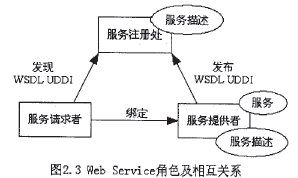
服务的平台。

(2)服务请求者

从企业的角度来看它是一个寻求一定服务功能的请求者，从整个体系的角度来看它也是～个寻找和调用服务的应用程序。服务请求者可以是浏览器(由入操作)，也可以是一个没有用户界面的程序(如另外一个Web服务注册处它是服务提供者发布其服务描述的地方，同时也是服务请求者发现服务并且得到绑定信息(包含在服务描述之中)的场所。这种绑定可以是静

态的绑定(开发过程中)．也可以是动态的绑定(运行过程中)。对于静态的情况来说，服务注册处在整体的结构体系中是一个可选的角色，因为服务提供者可以将相关的服务描述直接送到服务请求处。同样，服务请求者亦可以从其它的地方得到服务描述，例如本地文件、FTP站点、Web站点、ADS(Advertisement and Discovery of Services)、DISCO(Discovery of Web

Service)等等。



。服务提供者把自己的服务发布

到服务注册处，而服务请求者到服务注册处查找自己感兴趣的服务，得到

相应的服务信息后，接着对该服务进行绑定，这样服务请求者就可以调用

服务提供者提供的服务了。从这个过程中我们可以看出，服务提供者和服

务请求者之间不是直接相连的，而是通过了服务注册处这个中间环节。这

样做可以满足动态绑定的要求，因为如果一个服务提供者想要加入到系统

中，它只要把自身的服务发布到服务注册处以供服务请求者调用就可以

了。而从服务请求者的角度来说，它可以方便地调用已经在服务注册处注

册的服务提供者提供的服务。

Web Service的主要目标是跨平台的可互操作性。为了实现这一目标，Web Service 完全基于XML（可扩展标记语言）、XSD（XML Schema）等独立于平台、独立于软件供应商的标准，是创建可互操作的、分布式应用程序的新平台。因此使用Web Service有许多优点:

1、跨防火墙的通信

如果应用程序有成千上万的用户，而且分布在世界各地，那么客户端和服务器之间的通信将是一个棘手的问题。因为客户端和服务器之间通常会有防火墙或者代理服务器。传统的做法是，选择用浏览器作为客户端，写下一大堆ASP页面，把应用程序的中间层暴露给最终用户。这样做的结果是开发难度大，程序很难维护。要是客户端代码不再如此依赖于HTML表单，客户端的编程就简单多了。如果中间层组件换成Web Service的话，就可以从用户界面直接调用中间层组件，从而省掉建立ASP页面的那一步。要调用Web Service，可以直接使用Microsoft SOAP Toolkit或.net这样的SOAP客户端，也可以使用自己开发的SOAP客户端，然后把它和应用程序连接起来。不仅缩短了开发周期，还减少了代码复杂度，并能够增强应用程序的可维护性。同时，应用程序也不再需要在每次调用中间层组件时，都跳转到相应的“结果页”。

2、应用程序集成

企业级的应用程序开发者都知道，企业里经常都要把用不同语言写成的、在不同平台上运行的各种程序集成起来，而这种集成将花费很大的开发力量。应用程序经常需要从运行的一台主机上的程序中获取数据；或者把数据发送到主机或其它平台应用程序中去。即使在同一个平台上，不同软件厂商生产的各种软件也常常需要集成起来。通过Web Service，应用程序可以用标准的方法把功能和数据“暴露”出来，供其它应用程序使用。

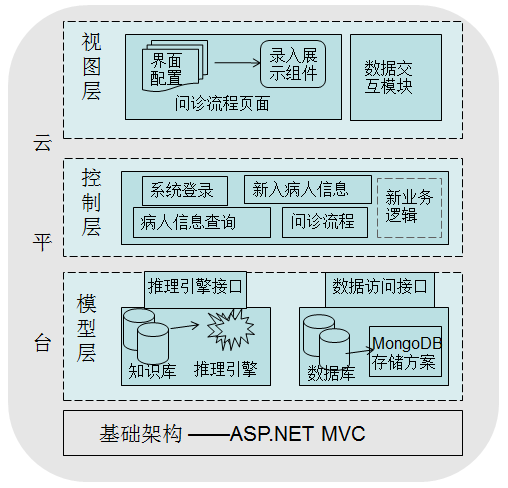
XML Web services 提供了在松耦合环境中使用标准协议（HTTP、XML、SOAP 和 WSDL）交换消息的能力。消息可以是结构化的、带类型的，也可以是松散定义的。

4、软件和数据重用

Web Service在允许重用代码的同时，可以重用代码背后的数据。使用Web Service，再也不必像以前那样，要先从第三方购买、安装软件组件，再从应用程序中调用这些组件；只需要直接调用远端的Web Service就可以了。另一种软件重用的情况是，把好几个应用程序的功能集成起来，通过Web Service “暴露”出来，就可以非常容易地把所有这些功能都集成到你的门户站点中，为用户提供一个统一的、友好的界面。可以在应用程序中使用第三方的Web Service 提供的功能，也可以把自己的应用程序功能通过Web Service 提供给别人。两种情况下，都可以重用代码和代码背后的数据

### 2.2 系统框架设计

应用以上相关技术，设计系统的整体框架图如下所示。



#### 2.2.1 架构模式选择

MVC的英文是Model-View-Controller，概念上将视图从流程控制、业务逻辑独立开来，并定义了相互间作用的机制，使得各个模块的开发相对独立，这样一个应用被分成三个层——模型层、视图层、控制层。

1．Model

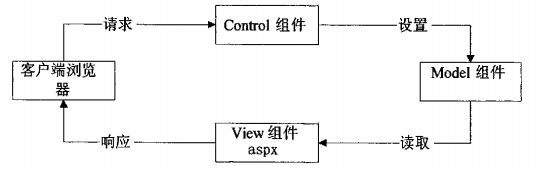
事务逻辑模块，通常是系统的核心部分，它包含数据操作和业务规则。它表示的是解决方案空间的真正的逻辑。模型的构成与具体的应用问题紧密相关。通常模型包括数据访问、业务逻辑和业务规则。

2．View

用户视图模块，这是用户界面部分。此模块承担连接用户和应用程序之间的接口角色，是一个系统中与用户关系最为密切的部分，也是需求变化最容易发生的部分。一方面，View模块将用户的请求传递给应用程序，触发应用程序对用户请求的处理逻辑；另一方面，View模块将请求的处理结果返回给用户。

3．Controller

流程控制模块，这是前面两者的接口。控制器根据用户的操作判断所要执行的业务逻辑，关主要用于接收和转发用户请求。通常，从视图接收到用户请求后根据用户提供的业务信息传递给相应的模型处理，再将结果结合相应的视图返回展示给用户。

（图要改

综上，MVC的工作流程是：

(1)控制器接收用户请求并调用相应的模型进行处理；

(2)模型处理用户请求并返回数据；

(3)控制器利用返回的数据渲染相应的视图展示给用户。

采用MVC模式进行系统框架的开发，可以降低数据描述和应用操作的耦合度，使系统结构清晰，各模块内的组件可复用性易于实现。而且各个部分的分离使得系统的维护性和扩展性得到极大的提高。

#### 2.2.2 基于Web Service的推理引擎模块

（1）功能分析

推理引擎部分是知识规则应用于问题求解的载体，根据系统输入数据的当前内容，利用知识库的内容，按照一定的推理策略，去解决当前的问题。由于系统需要解决的不是使用单个推理引擎解决某种疾病的诊断问题，因此推理引擎模块需要提供多种推理方式的统一的推理引擎接口。然而由于推理方式种类繁多，已有的实现方式多样，基于各种编程语言和平台。而基于web service 的推理引擎模块将推理的内部实现封装起来，以服务的形式向上层提供推理功能。也就是说，在保存接口一致的前提下，推理方法的调换或者更新都可以轻松实现，为简化系统后续的维护和更新的打下了基础。此外，web service的使用和管理都独立于系统其它部分，而且与开发语言无关，可以满足开发人员的协同工作和多样性的需求。

（2）模块内部流程设计

流程图及解释

（3）服务开发流程设计

发布- - 注册---动态调用

#### 2.2.3 基于MongoDB的数据访问模块

（1）功能分析

医疗数据由于本身的复杂性，往往不是典型的关系型数据，数据类型繁杂且具有较多的层级结构。而关系型数据库的中的“行（row）”概念划定了严格的字段，MongoDB的“文档（Document）”模型灵活可变，通过内嵌文档或者数组的方式可支持复杂的层次关系。另一方面，就临床医疗数据的来说，需要对于医疗数据进行有效的检索及统计分析。MongoDB在文档的查询上，支持通用的辅助索引和复合索引，能够提供快速方便的查询；而统计分析方面，MongoDB支持MapReduce和其他聚合工具，对于大量数据的统计分析及数据挖掘提供良好的编程基础。

（2）数据库设计

数据获取接口

数据结构

四层



存储层：

通信层：

服务层：

接口层：

#### 2.2.4 基于可视化组件的录入展示模块

（1）功能分析

在疾病诊断过程中，医生是对于病人的信息进行综合处理，这个过程是由三个阶段的循环来表示的：观察（Observation）,诊断（Diagnosis）和治疗（Therapy），如下图所示。在过程中，医生通过回忆或搜索相关的医学知识和经验，向病人询问他/她相关的问题，如临床症状、相关病史、疾病诱发原因、疾病发作情况以及病患家族史等，搜集信息后集合疾病的知识进行诊断推理，并得出一个结论，然后给出治疗的方案。

在这个过程中，如果在问诊过程中，对于病人的相关信息了解越充分，关于疾病的不确定性会越小，因此诊断决策的准确性往往取决于问诊过程中信息的获取程度。然而不同疾病对于病人的信息获取有着相当大的差异，医生如果不了解需要获取的信息类型，很难得出正确的诊断结论。对于系统来说，也就是疾病问诊流程部分的数据获取必须随着疾病的需求而变化，否则不能起到辅助决策的作用。因此，针对疾病流程的动态性，采用可视化组件将可变的界面存为配置文件，并通过数据交互模块完成数据的传输功能。

（2）数据交互过程设计

图

流程说明

采用Ajax技术，同步调用，将数据传往服务端

* SetDataIntoJson: 将界面数据从组件接口取出转换成Json数据格式
* GetDataFromJson : 从服务器传回的Json解析得到数据，并通过组件接口显示数据。

### 2.3 系统框架实现

#### 2.3.1 控制层

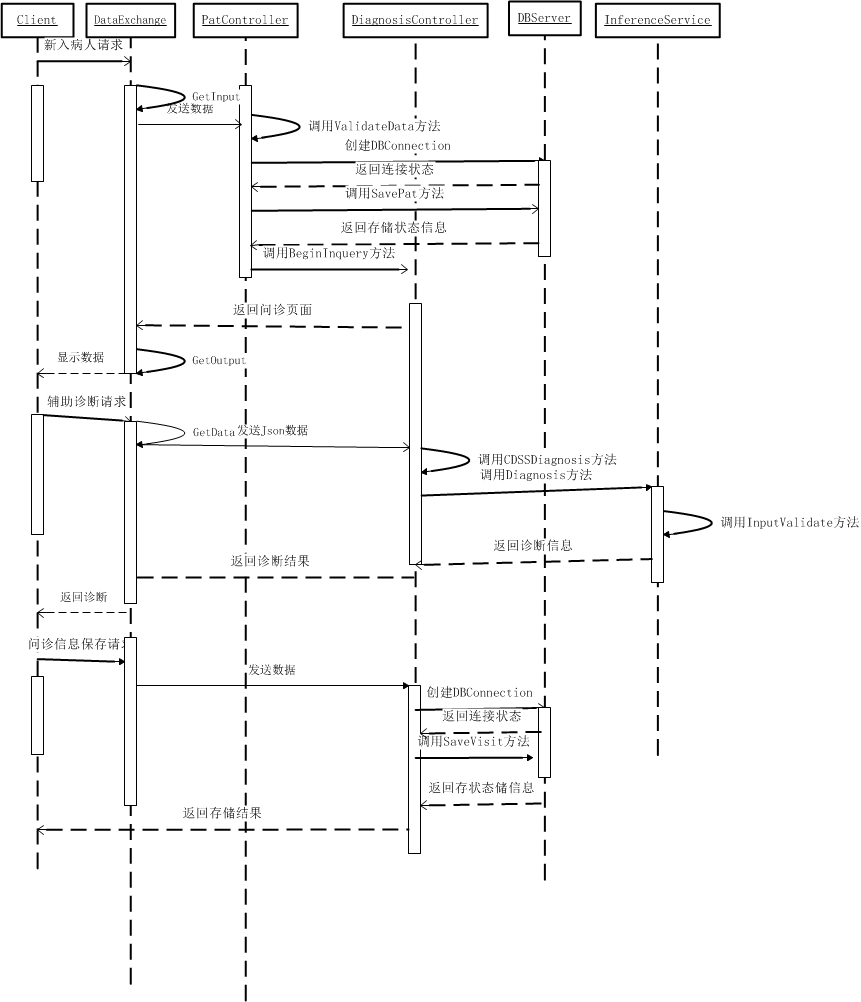
框架的控制器 Controller 是整个 Web 应用程序的控制中心。它承担的主要功能如下：接收浏览器客户端的处理请求，根据用户的请求和需要执行的业务逻辑进行处理，然后获取业务逻辑的处理结果，并将返回的结果组织成Json 文件的形式，通过视图层将其进行转换并显示给最终用户。

首先根据业务需求实现了以下模块

* AccountController
* PatInforController
* VisitRecordController
* DiagnosisController

本系统框架包含的主要业务逻辑如下：

（1）新入病人的问诊流程



（2）已就诊过的病人信息查询

（3）

#### 2.3.2 视图层

视图层是主要负责为浏览器客户端提供动态页面显示，视图接收用户输入的数据，并能将最后的数据显示给用户，但它并不接收任何业务逻辑处理。在本系统开发框架中，使用 数据交互接口来转换控制层Controller返回的Json形式的结果集并最终展示给用户

本文使用HTML和CSS开发了以下视图模板，为控制层提供人机交互接口。

1. 系统登录视图模板

系统初始登录页面，提供权限验证功能，如图

1. 病人信息查询视图模板

新录病人信息或者查询已就诊病人信息，如图

1. 病人问诊记录查询视图模板

根据问诊日期显示病人历次就诊记录信息，如图

1. 问诊流程视图模板

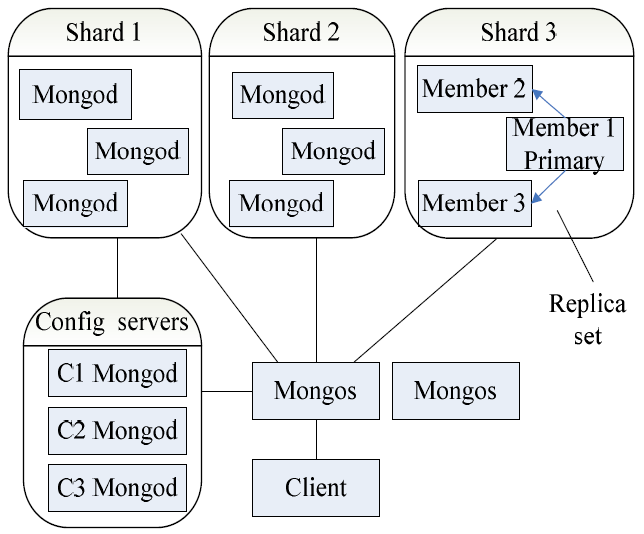
提供问诊向导的功能，结合可视化组件可达到问诊流程可配置的效果，如图

#### 2.3.3 模型层

模型是应用程序的主体部分，它被用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。一个模型可以被多个视图提供据，所以提高了应用的可重用性。本系统框架中模型层主要分为两部分---数据操作和推理服务

##### 2.3.3.1 数据持久化

（1）分布式集群搭建

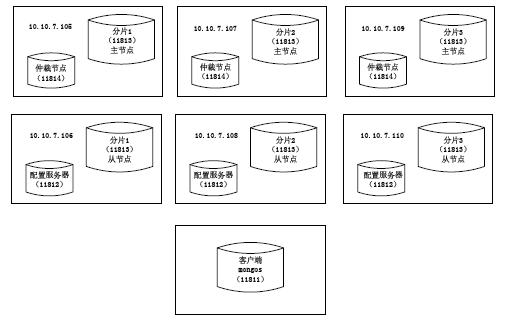


1、分片：一个分片为一组mongod，通常一组包括两台mongod，这两台mongod互为主从模式。而且存储的数据相同，互为数据备份。一个完整的数据按照一定的顺序进行分割，分别存储到不同的分片中。每个分片上的数据都是一个范围内的连续数据，所以支持指定分片的范围查询。当数据块超过指定容量大小时，数据块将会分成两块，当某个分片的数据过多，或者新的分片加入集群，数据将自动迁移，均衡到整个集群系统中。

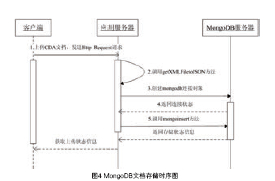
2、mongos：mongos作为一个路由处理节点，类似集群系统的控制中心，负责路由和集群的协调工作，使得集群成为一个整体。mongos启动的时候，从配置服务器获取分片的信息，当客户端请求存储服务时，mongos将请求路由到各个分片上，各个分片操作完毕之后，再将汇总结果返回给客户端。mongos路由节点可以运行在分片上，也可以运行在客户端上，一般而言，将mongos路由节点运行在分片上可以减少网络消耗。

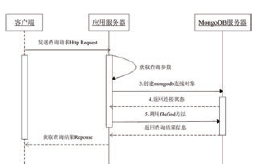
3、config server：存储集群的信息，包括分片和块数据信息。主要存储块数据信息，每个config server上都有一份所有块数据信息的拷贝，以保证每台config server上的数据的一致性。 4、shard key：为了分割数据集，需要制定分片key的格式，类似于用于索引的key格式，通常由一个或多个字段组成以分发数据，比如： { name : 1 } { \_id : 1 } { lastname : 1, firstname : 1 } { tag : 1, timestamp : -1 } mongoDB的分片为有序存储(1为升序，-1为降序)，shard key相邻的数据通常会存在同一台服务（数据块）上。

MongoDB服务器分布式部署有很多种方式。在部署的时候，可以将每个config、mongos、mongod都单独运行在独立的服务器上。也可以将config、mongos与mongod服务器共存。后者更为常用，节省了服务器资源。下图为物理机共享的集群。config和mongos与mongod共享同一台服务器，不需要为它们单独配置服务器[34]。 在实验室环境下，在服务器中虚拟了6台虚拟机，每两台虚拟机组成一个分片。一共有三个分片。每台分片有一个replica（副本集），每个replica由两台虚拟机组成，有一个主节点，一个从节点。两个节点的数据是一样的，形成数据备份。 实验环境如图4.2所示： 6台虚拟机 虚拟机配置：CPU 1Ghz，内存 1G，硬盘 100G 操作系统： Ubuntu 12.04.1 LTS x86\_64 MongoDB：version: 2.2.2 虚拟主机： mongo01： 10.10.7.105 mongo02： 10.10.7.106 mongo03： 10.10.7.107 mongo04： 10.10.7.108 mongo05： 10.10.7.109 mongo06： 10.10.7.110



（2）数据操作类实现





##### 2.3.3.2 推理服务

（1）推理流程

时序图

（2）推理引擎封装

目前系统框架内已包含两种推理方法

* 基于规则的推理实现

开源CLIPS引擎

* 基于数据的贝叶斯网络推理

weka

### 2.4专科疾病诊断决策支持系统构建方案

### 2.5小结

## 3.头痛诊断决策支持系统实现

### 3.1 系统需求分析

#### 3.1.1疾病特征分析

#### 3.1.2临床知识的特性

### 3.2 基于框架的系统开发流程

#### 3.2.1推理引擎选择

#### 3.2.2 数据模型设计

信息类表

#### 3.2.3 问诊界面配置

#### 3.2.4 数据交互层实现

### 3.3 系统实现

### 3.4小结

## 4.老年痴呆症诊断决策支持系统实现

### 4.1 系统需求分析

#### 4.1.1疾病特征

#### 4.1.2临床知识的特性

### 4.2 基于框架的系统开发流程

#### 4.2.1推理引擎选择

#### 4.2.2 数据模型设计

#### 4.2.3 问诊界面配置

#### 4.2.4 数据交互层实现

### 4.3 系统实现

### 4.4小结

## 5.总结与展望

### 5.1 总结

### 5.2 展望

1. 赵志威: **我国社区医疗现状及推行首诊制的必要性**. *中国中医药咨讯* 2010, **2**(28).

2. 黎友隆, 林少东, 罗雅霞: **社区医疗服务的发展策略研究**. *经济研究导刊* 2013(8):164-168.

3. 刘尚辉, 曾文: **建立城乡社区疾病规范化诊疗智能决策知识系统的构想及探讨**. *中国全科医学* 2011, **14**(22).

4. 刘佳, 冯泽永: **社区首诊制的实施困境分析及对策研究**. *中国全科医学* 2012, **7**:006.

5. 赖光强, 王跃平, 陈建, 张炜, 陈皞璘: **深圳新型社区首诊制实施效果分析与思考**. *中国全科医学* 2009, **12**(2):202-203.

6. 田翠环, 胡燕生: **电子病案信息与社区医疗共享**. *中国病案* 2011, **12**(6):48-48.

7. 谢礼琼, 李林平, 王明霞: **借鉴英国卫生服务体系探讨我国社区卫生体系建设**. *重庆医学* 2010, **3**(5):620-622.

8. Osheroff JA, Teich JM, Middleton B, Steen EB, Wright A, Detmer DE: **A roadmap for national action on clinical decision support**. *Journal of the American medical informatics association* 2007, **14**(2):141-145.

9. McCoy AB, Melton GB, Wright A, Sittig DF: **Clinical Decision Support for Colon and Rectal Surgery: An Overview**. *Clinics in Colon and Rectal Surgery* 2013, **26**(01):023-030.

10. Friedman CP, Elstein AS, Wolf FM, Murphy GC, Franz TM, Heckerling PS, Fine PL, Miller TM, Abraham V: **Enhancement of clinicians' diagnostic reasoning by computer-based consultation**. *JAMA: the journal of the American Medical Association* 1999, **282**(19):1851-1856.

11. Lincoln MJ, Turner C, Haug P, Williamson J, Jessen S, Cundick R, Cundick K, Warner H: **Iliad's role in the generalization of learning across a medical domain**. In: *Proceedings of the Annual Symposium on Computer Application in Medical Care: 1992*. American Medical Informatics Association: 174.