

М.В. БУРЦЕВ, НТУ "ХПИ", Харьков,

А.И. ПОВОРОЗНЮК, д-р техн. наук, НТУ "ХПИ", Харьков

АРХИТЕКТУРА СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В МЕДИЦИНЕ, ОСНОВАННОЙ НА КОМБИНИРОВАННОМ РЕШАЮЩЕМ ПРАВИЛЕ

Рассмотрена структура системы поддержки принятия решений в медицине, основанной на комбинированном решающем правиле. Описана архитектура разрабатываемого программного обеспечения. Рассмотрены схемы развертывания системы. Ил.: 2. Библиогр.: 10 назв.

Ключевые слова: система поддержки принятия решений, комбинированное решающее правило, архитектура программного обеспечения.

Постановка проблемы и анализ литературы. Успешность реализации информационной системы во многом зависит от выбора целевой платформы разработки, а также правильно спроектированной архитектуры. В [1] рассмотрена реализация комбинированного решающего правила (КРП) [2] для системы поддержки принятия решений (СППР) в медицине, обоснован выбор Java [3 – 4] в качестве основной платформы: обеспечивается возможность развертывания системы в различных аппаратно-программных средах, неpropriетарный характер платформы, наличие множества открытых библиотек, поддерживаемых сообществом разработчиков.

При проектировании системы необходимо изначально заложить архитектурные принципы, которые в будущем обеспечат масштабируемость, гибкость, а также простоту сопровождения системы. Для обеспечения данных качеств, в настоящее время, широко применяются шаблоны проектирования [5].

Обычно шаблон не является законченным образцом, который может быть прямо преобразован в код; это лишь пример решения задачи, который можно использовать в различных ситуациях. Объектно-ориентированные шаблоны показывают отношения и взаимодействия между классами или объектами, без определения того, какие конечные классы или объекты приложения будут использоваться [6]. Главная польза каждого отдельного шаблона состоит в том, что он описывает решение целого класса абстрактных проблем. Также тот факт, что каждый шаблон имеет свое имя, облегчает дискуссию об абстрактных структурах данных между разработчиками, так как они могут ссылаться на известные шаблоны. Таким образом, за счёт шаблонов производится унификация терминологии, названий модулей и элементов проекта.

Целью статьи является обоснование методики реализации системы поддержки принятия решений в медицине, основанной на комбинированном решающем правиле, а также описание разработанной архитектуры системы.

Структура СППР. В структуре разрабатываемой СППР можно выделить три основных модуля (рис. 1): модуль взаимодействия с пользователем, базу данных, включающую в себя базу знаний и модуль построения знаний.

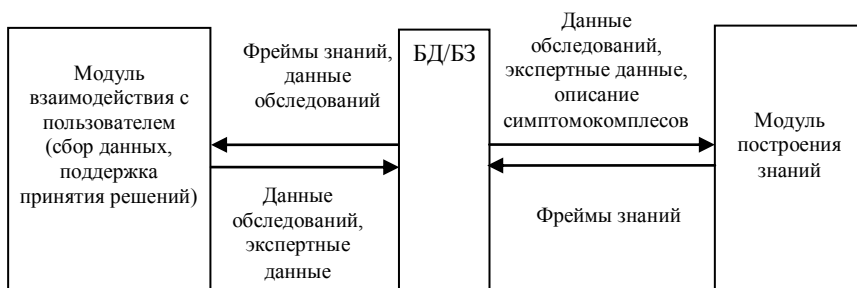


Рис. 1. Структурная схема СППР

Модуль взаимодействия с пользователем представлен графическим интерфейсом пользователя и позволяет осуществлять сбор данных обследований, административных данных, а также данных предоставляемых экспертами, которые используются при формировании знаний системы.

Для хранения данных и фреймов знаний системы используется реляционная база данных (БД).

Модуль построения знаний отвечает за формирование фреймов знаний, представленных иерархической структурой диагнозов [7], имеющей вид двоичного дерева, каждый простой узел которого содержит конечный диагноз, а составные – две группы диагнозов, относительно которых осуществляется диагностика, а также последовательности наборов диагностически значимых интервалов численных признаков [8] и функции принадлежности, используемые КРП, заданные экспертами [9].

Архитектура СППР. При проектировании системы необходимо четко разграничить уровни данных, логики и представления разрабатываемой системы. Подобное разделение соответствует шаблону модель-представление-контроллер (MVC pattern) в архитектурном плане. Архитектура разрабатываемой системы представлена на рис. 2.

На уровне данных выделяются сущности (entities), необходимые для решения задачи диагностики (как административные, так и сущности, описывающие признаки, диагнозы и т.д.). Сущности содержат в себе исключительно данные. Менеджеры сущностей представляют собой реализацию шаблона объекта доступа к данным (DAO pattern), и предоставляют возможность выполнения CRUD-операций над имеющимися данными: осуществляют выборку, изменение, добавление и удаление экземпляров сущностей. Менеджеры сущностей взаимодействуют с уровнем логики приложения, при этом сами бизнес-логики не содержат; являются единственным источником данных для всех сервисов системы.

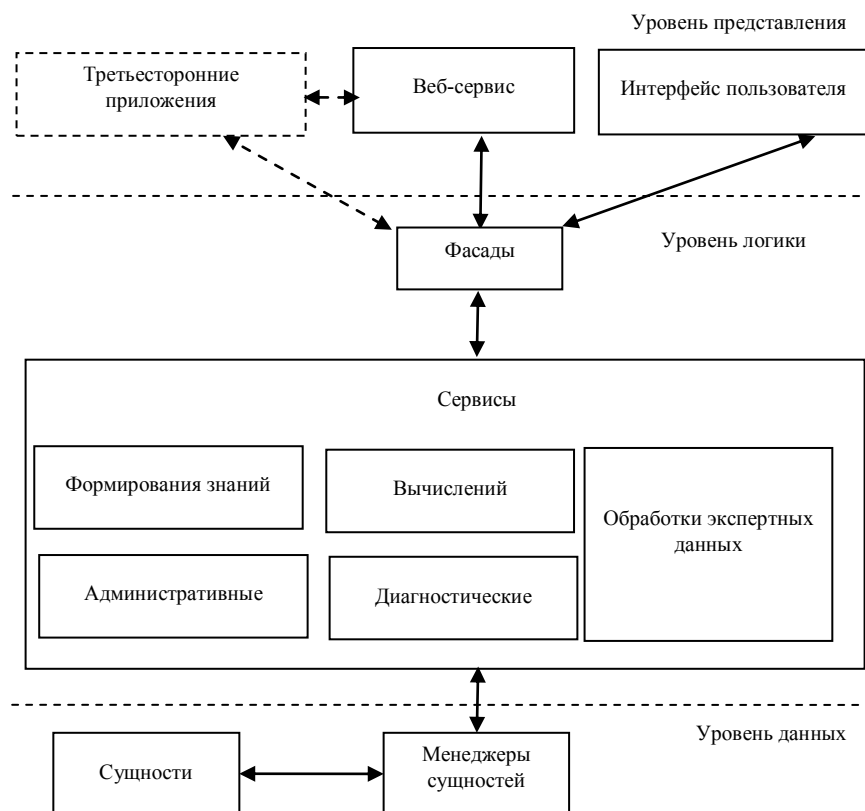


Рис. 2. Архитектура СПИР

Связь данных БД и сущностей приложения осуществляется при помощи механизма объектно-реляционного связывания (ORM). Для

этого предполагается использование библиотеки Hibernate [10], которая позволяет описать связи полей сущностей с колонками таблиц БД (с помощью аннотаций, либо XML-описания), при этом исчезает необходимость в написании большого объема JDBC-кода для получения данных из БД.

Уровень логики включает в себя сервисы системы, содержащие всю бизнес-логику приложения. Административные сервисы отвечают за обработку данных о пациентах, сотрудниках, ведение соответствующей медицинской документации. Сервисы вычислений предоставляют доступ к различным алгоритмам: формирование диагностически значимых интервалов признаков, генетические алгоритмы и т.д. Сервисы формирования знаний являются реализацией шаблона строитель (Builder pattern), осуществляют построение фреймов знаний системы, структура которых рассмотрена выше. Сервисы обработки экспертных данных осуществляют обработку данных, предоставленных экспертами (функции принадлежности, описание структуры симптомокомплексов, веса признаков и др.) и подготавливают метаданные для работы сервисов формирования знаний. Диагностические сервисы, используя знания системы, осуществляют поддержку принятия решений: на основании входных данных о результатах обследования предлагается некоторое множество диагнозов, соответствующих состоянию диагностируемого.

Фасады (Facade pattern) обеспечивают представление ядра системы во вне (API для интерфейса пользователя и третьесторонних приложений), скрывая от внешнего мира всю внутреннюю реализацию и структуру системы. Каждый из фасадов является оберткой для одного, либо нескольких сервисов, при этом может содержать дополнительную логику (например, валидацию входных данных).

Уровень представления включает графический интерфейс пользователя (GUI) и веб-сервис. GUI является самостоятельным приложением, реализованным в соответствии с требованиями шаблона MVC. GUI использует API ядра системы, предоставляемые фасадами.

Взаимодействие с третьесторонними приложениями может осуществляться как непосредственно через фасады (в том случае, если третьестороннее приложение является Java-приложением, то возможен прямой вызов методов фасадов, используя механизм RMI), так и через веб-сервис.

Веб-сервис является оберткой для всех фасадов системы, предоставляет возможность общаться с ядром системы с помощью обмена SOAP-сообщениями.

Развертывание системы. Простейший вариант представляет собой развертывание всех структурных элементов на одном сервере. Такой

подход позволяет сэкономить на затратах, связанных с приобретением аппаратных ресурсов, но не предполагает высокой нагрузки на систему. Является оптимальным для небольших организаций.

Для обеспечения более высокой производительности и обслуживания крупных организаций необходимо выполнить размещение модулей системы на различных серверах (GUI, БД/БЗ, модуль построения знаний). Если же желаемая производительность не достигнута, то необходимо выполнить кластеризацию отдельных модулей, несущих наибольшую нагрузку, а также конфигурирование балансировщика нагрузки (loadbalancer).

При развертывании системы возможен отказ от приобретения физических серверов и выполнение развертывания в облаке (Amazon EC2, Jelastic), если позволяет бюджет организации.

Выводы. Обоснован подход к реализации системы поддержки принятия решений в медицине, рассмотрены ее основные модули. Описана архитектура разрабатываемой системы. Рассмотрены схемы развертывания системы.

Список литературы: 1. *Бурцев М.В.* Программная реализация комбинированного решающего правила для задач медицинской диагностики / *М.В. Бурцев, А.И. Поворознюк* // Вісник НТУ "ХПІ". Серія: Інформатика і моделювання. – Харків: НТУ "ХПІ". – 2010. – № 21. – С. 11–16. 2. *Бурцев М.В.* Синтез комбинированного решающего правила в задаче медицинской диагностики / *М.В. Бурцев, А.И. Поворознюк* // Вісник НТУ "ХПІ". Серія: Інформатика і моделювання. – Харків: НТУ "ХПІ". – 2009. – № 43. – С. 27 – 33. 3. *Эккель Б.* Философия Java / *Б. Эккель* – СПб.: Питер, 2009. – 640 с. 4. *Шилдт Г.* Полный справочник по Java / *Г. Шилдт*. – М.: ООО "И.Д. Вильямс", 2009. – 1040 с. 5. *Бек К.* Шаблоны реализации корпоративных приложений / *К. Бек*. – М.: ООО "И.Д. Вильямс", 2008. – 176 с. 6. *Стелтинг С.* Применение шаблонов Java / *С. Стелтинг., О. Маасен* – М.: ООО "И.Д. Вильямс", 2002. – 576 с. 7. *Бурцев М.В.* Построение иерархической структуры диагнозов для комбинированного решающего правила в компьютерных системах медицинской диагностики / *М.В. Бурцев, А.И. Поворознюк* // Вісник НТУ "ХПІ". Серія: Інформатика і моделювання. – Харків: НТУ "ХПІ", 2011. – С. 29–34. 8. *Поворознюк А.И.* Формирование диагностических интервалов численных признаков при дифференциальной диагностике / *А.И. Поворознюк* // Вісник Хмельницького національного університету. – Хмельницький: ХНУ, 2007. – № 3. – Т. 1. – С. 106–109. 9. *Бурцев М.В.* Выбор функций принадлежности для описания симптомокомплексов вкомбинированном решающем правиле / *М.В. Бурцев, А.И. Поворознюк* // Вісник НТУ "ХПІ". Серія: Інформатика і моделювання. – Харків: НТУ "ХПІ". – 2010. – № 31. – С. 10–15. 10. *Хемрадджани А.* Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse / *А. Хемрадджани*. – М.: ООО "И.Д. Вильямс", 2008. – 352 с.

УДК 681.3

Архітектура системи підтримки прийняття рішень в медицині, заснованої на комбінованому вирішальному правилі / Бурцев М.В., Поворознюк А.І. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2012. – № 38. – С. 26 – 31.

Розглянуто структуру системи підтримки прийняття рішень в медицині, заснованої на комбінованому вирішальному правилі. Описано архітектуру програмного забезпечення, що

розробляється. Розглянуто схеми розгортання системи. Лл.: 2. Бібліогр.: 10 назв.

Ключові слова: система підтримки прийняття рішень, комбіноване вирішальне правило, архітектура програмного забезпечення.

UDC 681.3

Architecture of medical support decision making system based on combined decision rule / Burtsev M.V., Povorozhuk A.I. // Herald of the National Technical University "KhPI". Subject issue: Information Science and Modeling. – Kharkov: NTU "KhPI". – 2012. – №. 38. – P. 26 – 31.

Medical support decision-making system structure is reviewed. Developing software architecture is described. System deployment schemes are reviewed. Figs.: 2. Refs.: 10 titles.

Keywords: support decision-making system, combined decision rule, software architecture.

Поступила в редакцію 09.06.2012